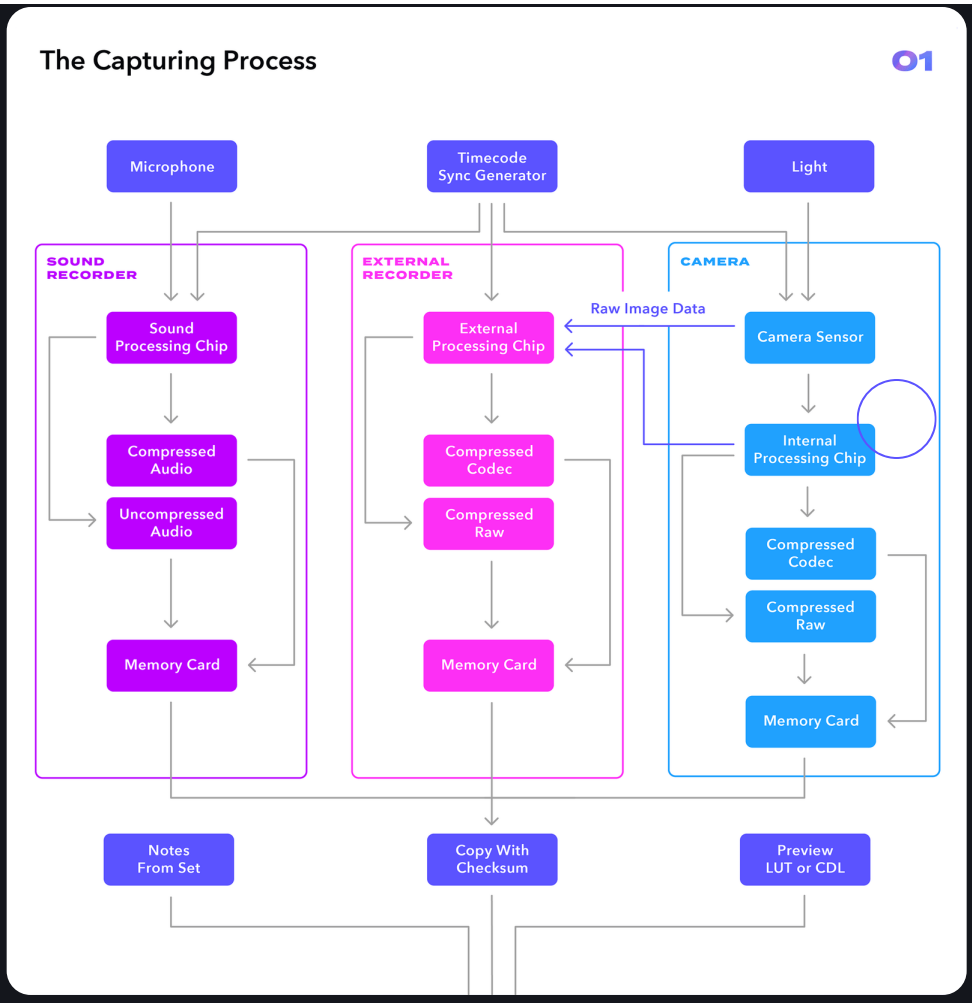
CHAPTER 01

# Capture

We dive into camera technology so that you can start your workflow off right, covering topics like cameras, codecs, and color spaces.

*9 minute read*




The filmmaking process begins long before the cameras start rolling, but that is where our journey starts, as we chart the factors and decisions that you will need to make throughout the post-production process.

## How to choose your capture codec

One of the first decisions to make is what codec to use, since most cameras offer many different options for how to record your video.

It's vital to consider your complete workflow at this point, because the camera and the codec that you choose now will have major implications for your entire workflow. In order to make your final decision about the camera and codec, you'll need to return to this section after you've read the rest of this guide, but we'll give you a good overview here.

What is a codec?

Codecs are a core technical consideration for every step of post-production, so it is critical that everyone who touches your workflow understands the basics.

A codec is the set of rules that tells computers and electronic equipment how to handle your media files, most notably digital video footage. The term codec is a shortening of the words *compressor-decompressor* or *coder-decoder*. As the name implies, codecs make video files smaller for storage, and then turn the compressed data back into a usable image when you need to use it again.

[Note: Codecs are not the same as *containers*. Containers are

the file types that actually store digital video data, and can be used in conjunction with many different codecs. Imagine containers as the box where you put data, and codecs as the instructions for packing and unpacking the box. H.264, DNxHD, and ProRes are codecs, while .movs, .mxfs are containers.]

We need codecs because uncompressed video files are gigantic. A single minute of uncompressed 4K footage measures in the dozens of gigabytes. So, you will probably never have the option of working with uncompressed footage (video without a codec) all the way through your workflow. It's just too large and too complex to work with in most cases.

Codecs solve this problem by reducing the size of your footage and making it easier to work with across the post-production pipeline. But just because the image is compressed doesn't mean it will look worse. While not all codecs are created equal, there are many high-end codecs you can choose from that still deliver stunning image quality. The right codec will make your footage much easier to manage, and you may not even be able to tell a difference from the original.

That said, you *will* be able to tell a difference if you choose the wrong codec for certain post-production processes. Using a codec with too much compression might make your desired color-correction unachievable, or make your VFX work look unrealistic. On the other hand, codecs with too little compression might make your video uneditable on normal computer hardware, and cause issues when transferring data between teams or facilities. Different codecs are good in different contexts, but there is no single codec that works for every use case.

Choosing codecs is one of the most important technical decisions you will make for your project. To a large extent, codecs determine what you can do with your footage in post

and how complex your workflow will be. It is usually much safer, faster, and cheaper to pick codecs that you know will fit neatly into a pre-planned/pre-existing workflow, than it is to improvise a new workflow around codecs that you have not used before. The last thing you want is to find out some or all of your footage is unusable because the codec isn't compatible with your software or hardware. As with everything in post-production, test all your codec choices before the project begins.

Of course, to pick codecs effectively, you need to understand how they impact the visual qualities of video. While the mathematics and computer science behind codecs are quite complicated, there are only a few fundamental concepts you need to know to choose the right codecs for your production.

## Bit Depth

First, you need to understand *bit depth* (also called *color depth*). This is basically the number of colors that make up an image. The most common bit depths for video are 8-bit (16 million colors), 10-bit (1 billion colors), and 12-bit (68 billion colors). Higher bit depth codecs can store more colors, which makes video appear smoother and subtler. In general, codecs with a deeper bit depth provide a higher quality image, but they also make for larger files. It is best to use a deep bit depth codec for video capture, color correction, and VFX.

Bit Depth

To illustrate bit depth further, imagine you want to draw a picture of a sunset, but all you have is the basic box of 16 crayons. Real life sunsets have a huge variety of colors, from blinding yellows and oranges, to faint reds

and purples. If all you have are those 16 crayons, you won't really be able to draw all those different shades. You'll still be able to draw a picture, but it won't look that great.

But what if you had the 32-crayon box? Well, you'd be able to use twice as many colors, but your picture still wouldn't look like the real deal. Now what if you had a box of 1024 crayons? At this point, you might actually be able to draw a pretty decent picture. You won't have all the colors you need to draw a photorealistic sunset, but you will do much better than you did with just 16 or 32 crayons. And if you keep adding more crayons to your box, you'll be able to draw a better and better picture.

This example might seem a bit simplistic, but it helps to understand how bit depth works. A codec's bit depth is basically the box of crayons it can use to draw images. The deeper the bit depth, the bigger the box of crayons (number of colors), and the more realistic image it can produce.

## How Bit Depth is Measured

To fully understand bit depth, we need to explore how digital imagery works in greater detail.

You probably already know that digital still and motion images are made up of pixels, and that each pixel mixes together three primary color channels (red, green, and blue) to make many different colors in the image.

What you may not know is how these colors are defined by a computer. Each of the three color channels has a

range of possible values it can be assigned, and this range is stored as a single number. What determines how big or small that number can be is the number of *bits* the computer uses to store it.

A bit is just a binary unit of information, digitally displayed as a 0 or a 1. To store increasingly complex information, computers need to use more bits per integer. So, a 1-bit integer can only have 2 values (0 or 1), but a 2-bit integer can have 4 values (00, 01, 10, and 11), and a 3-bit integer can have 8 values (000, 001, 010, 011, 100, 101, 110, and 111). For every extra bit in the integer, a digital value can store twice as much information.

By increasing the number of bits per primary color channel, computers can store more complex color information. If each primary color channel uses an 8-bit binary integer, that means there are 256 possible shades of red, green, and blue that can be used for every pixel. Mixing every single one of these shades together would make 16,777,216 different colors (256x256x256), from pure black (a value of 00000000 for each channel) to pure white (11111111 for each channel), and every shade in between.

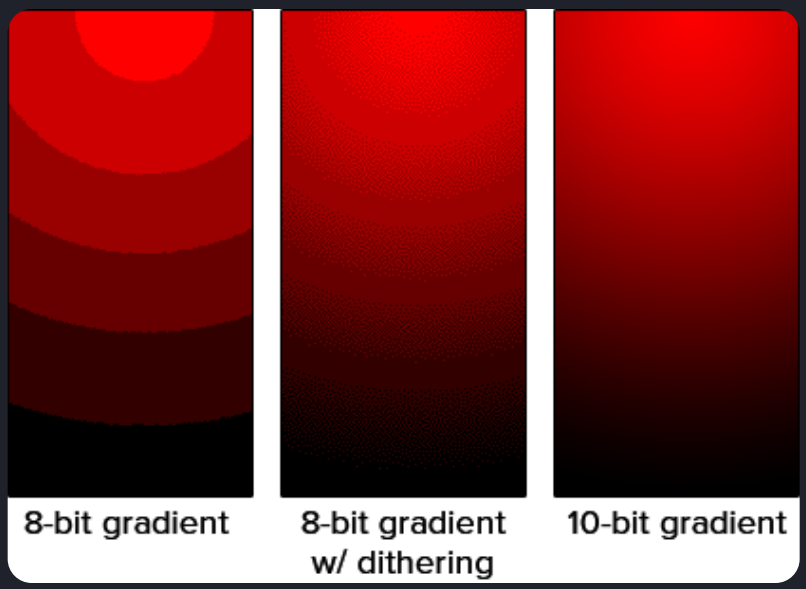
## 8-bit vs 10-bit

The most common bit depth for video is 8-bit. It has been the digital video standard for decades, both in television and film production. DVDs use 8 bits of color depth, as do Blu-rays, and most streamed content (for now). Even many cutting-edge cameras still default to 8-bit color when recording video. In fact, almost every digital video you've ever watched was 8-bit. We want to emphasize these points to demonstrate that video with

8 bits of color depth is technically capable, and can be used to tell incredible stories.

However, despite this, you should consider codecs with a bit depth higher than 8 bits if your workflow can handle it. Why? Because 8-bit codecs have some significant shortcomings that are important to understand.

The fundamental problem with 8-bit color depth is a lack of available colors. While 16.7 million colors might seem like a lot, remember that each primary color channel only has 256 unique shades to itself. That can make it difficult to reproduce subtle gradients between the same or similar colors.




This issue is called banding. It occurs because there is a relatively small number of steps between the brightest and the darkest shade of any given color. The lines in the image are artifacts left over from a computer trying to stretch too few available colors over too wide an information gradient. This problem is amplified even more in low contrast scenes, such as those that take place in dark rooms or under shadows.

The limitations of 8-bit codecs span the entire length of post-production processes, but are especially problematic for color correction and VFX. Strong color grading in an 8-bit codec often leaves noticeable artifacts in footage, simply because there aren't enough shades to make the fine-tuned adjustments you want. And chroma-keying (green screening) with an 8-bit codec can leave jagged or undefined edges, which makes VFX look bad.

Again, as the last 30+ years of digital video production has demonstrated, 8-bit color depth is usable for many projects. But, there are higher-quality options that you should consider for your workflow.

10-bit color depth is exactly like 8-bit color depth, except that each primary color channel has two extra bits for storing the color value. That means that instead of just 256 possible values per RGB channel, 10-bit yields 1024 possible values per RGB channel. That is a fourfold increase to 1,073,741,824 colors (1024x1024x1024).

10-bit has existed for high-end productions for decades, but has only become usable for everyday workflows recently as new cameras and software began to support it. It is now possible for even mid range DSLRs to shoot in 10-bit color depth, and to edit 10-bit codecs on standard editing workstations.

The advantages of 10-bit color depth almost always outweigh the extra storage and processing power you need to handle it, if you are doing any color correction or VFX. Your original camera files will be much higher quality, your color correction will be much more precise, and your VFX work will look much more realistic. Even if you will be delivering the video in an 8-bit codec, you should still consider mastering the project at 10 bits

because the intermediate steps will benefit.

## Chroma Subsampling

Next, you need to understand *chroma subsampling*. This is a method of reducing an image's size by throwing away the color data of some pixels and then replacing it with adjacent color. Chroma subsampling is expressed as a ratio, like 4:4:4 (no pixels lose color data), 4:2:2 (50% of pixels lose color data), and 4:2:0 (75% of pixels lose color data). This can save a tremendous amount of space, and in many situations will not be noticeable to viewers because the human eye is much more sensitive to brightness than to color changes.

However, you need to be aware of how much different codecs subsample color. Too much can visibly degrade image quality, and will make it difficult to manipulate the image in certain ways. In general, you should pick a codec with as little chroma subsampling as possible for image capture. That will give your original camera files lots of color data, which will make color correction and VFX work much easier. If you shoot a greenscreen shot in a 4:2:0 codec, for instance, you will be in a lot of trouble.

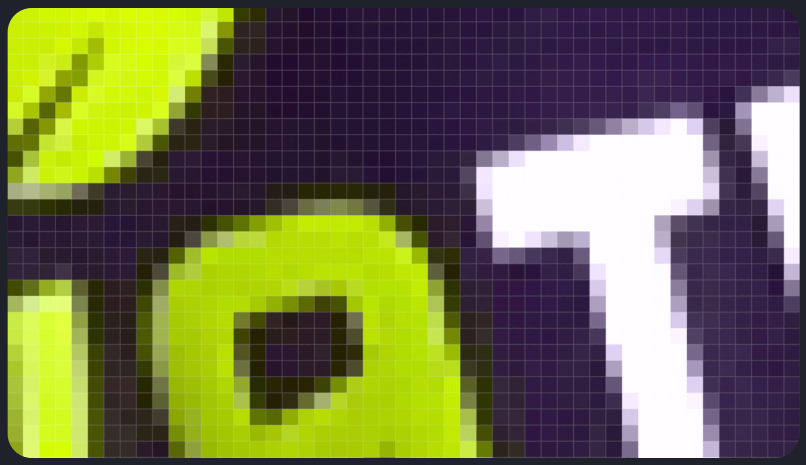
Chroma subsampling

If you've clicked this subsection you're probably wondering something like "how can an image lose 75% of its color and still look good?" It's an understandable question, so here we'll explore chroma subsampling in finer detail.

As previously mentioned, our eyes are more sensitive to

changes in luma (brightness) information than they are to chroma (color) information. Knowing this, engineers came up with a clever scheme for reducing image data size. If some pixels in an image discarded their color data, then more storage space could be devoted to things like resolution or framerate.

Of course, you can't just throw away color in an image. People would notice. So, chroma subsampling doesn't just get rid of colors, it shares chroma values between pixels. If a pixel loses its color information, it borrows some from another pixel next to it. This doesn't mean the pixels look the exact same, because they both retain their luma information. So, since most pixels vary at least a small amount in their recorded brightness, they will still appear slightly different, which preserves visual detail to the viewer.



The dark green color visible in the edge pixels of the letter is not actually present on real life object. It is merely a result of the chroma subsampling removing the chroma data of the blue pixels and replacing it with the chroma of the green pixels, but with the same luma information. There is some data loss, obviously, but in normal viewing conditions (when you're not looking at massively zoomed in pixels), this will be hardly noticeable.

So how exactly does this work?

## Chroma Subsampling Ratios

Chroma Subsampling is usually expressed in a three-part ratio, made up of the components *J*:*a*:*b*.

[Note: You might occasionally see a four-part ratio. In these instances, the fourth number represents the alpha (transparency) sample, and it will match the first number in the ratio. This annotation is less common, however.]

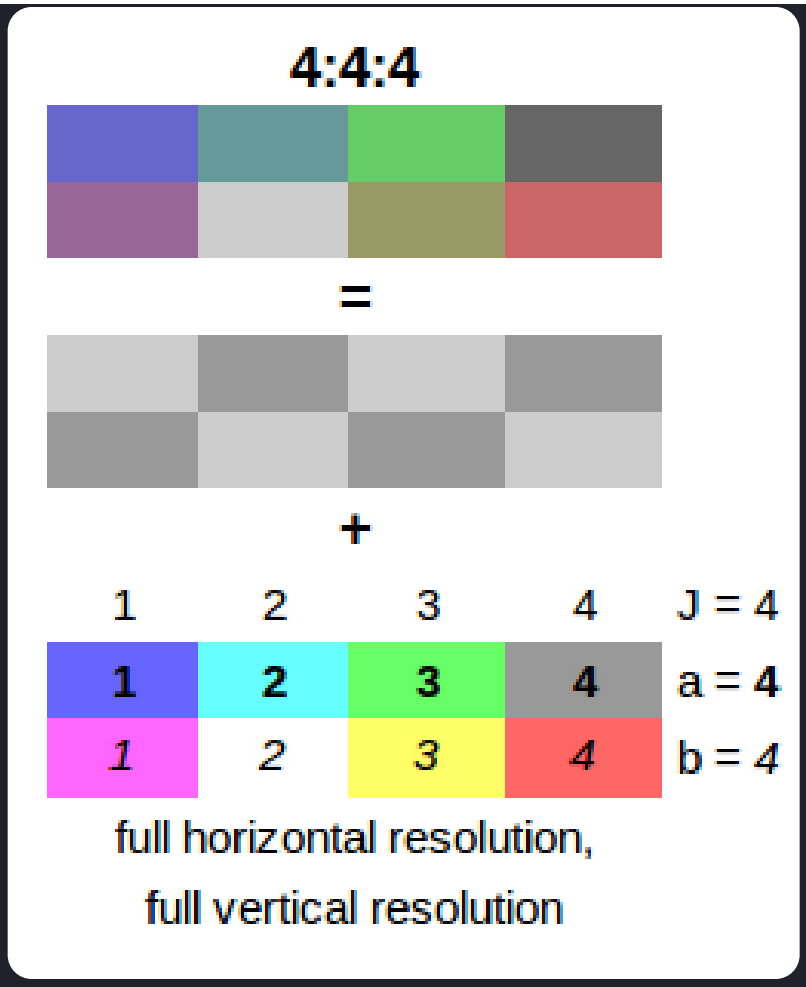
The *J* component is an indicator of how many pixels wide the compression reference block is. This is also a measure of how many luma samples there are in each row.

The *a* component is an indicator of how many samples of chroma are in the first row of the reference block.

The *b* component is an indicator of how many samples of chroma are in the second row of the reference block.

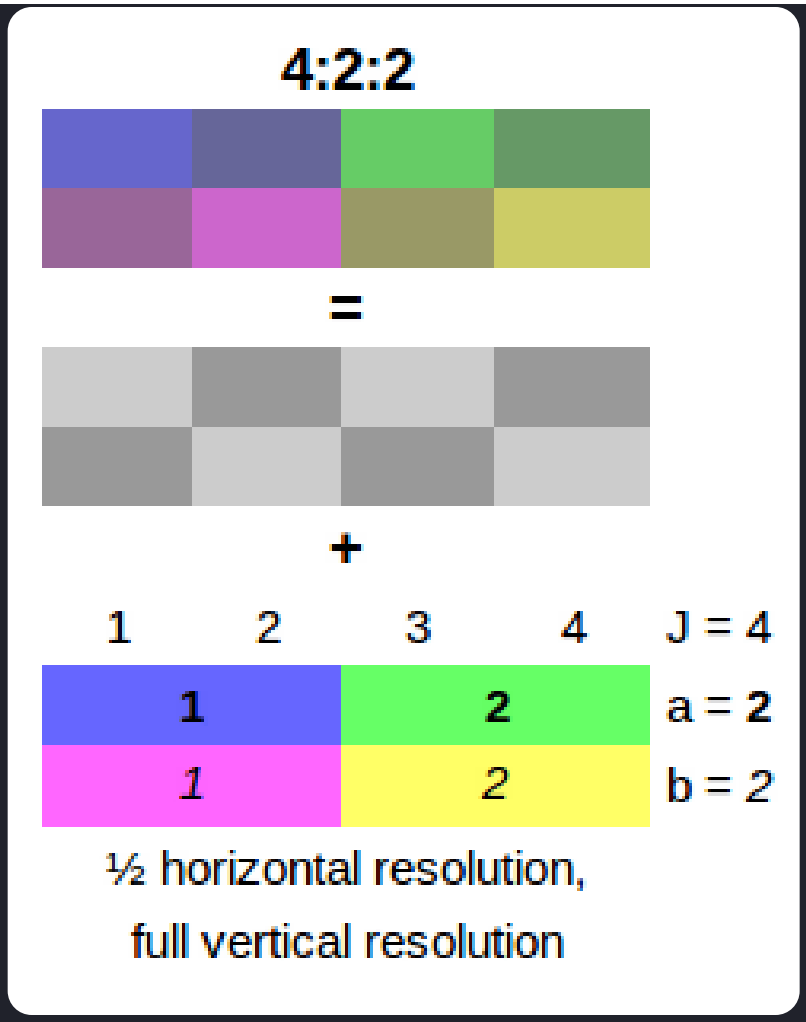
There are many different possible (and theoretical) chroma subsampling schemes, but we'll only cover the three most common: 4:4:4, 4:2:2, and 4:2:0.

Technically 4:4:4 means no chroma subsampling is taking place. Every pixels stores its own color information, so 100% of captured color information is available in the image data.



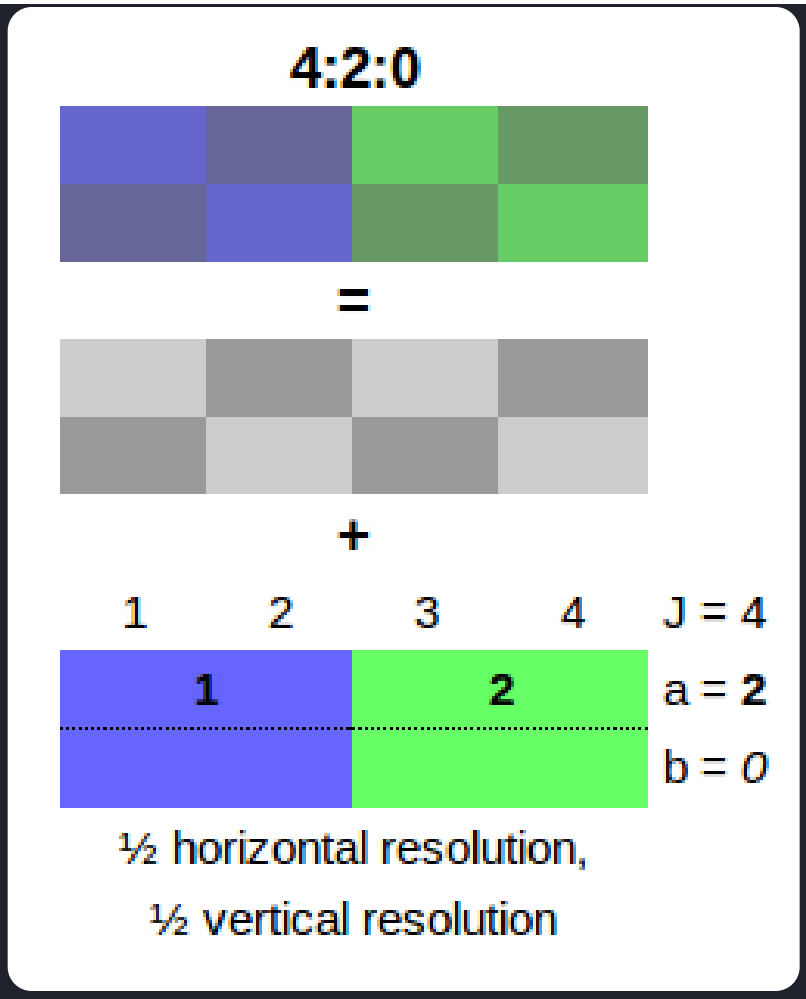

This scheme is used in high-end film scanners and top-tier cinematic post-production workflows, where maximum image quality is necessary.

In this scheme, every two pixels share chroma data. This means that 50% of all captured color information is replaced, which reduces the required bitrate by approximately one-third. However, this does not yield much visual difference.




Many high-end digital video formats and interfaces use this scheme, including the popular ProRes 422. As cameras capable of 10-bit image recording have entered the market, codecs that utilize 4:2:2 chroma subsampling have shifted from being a coveted, rare feature, to a standard for many professional productions.

For 4:2:0 chroma subsampling, only two pixels in the first row keep their chroma information, while the rest (two in the first row, and all four in the second row) copy these chroma values. This scheme replaces 75% of all captured chroma data, but allows for a 50% data savings in the bitstream.




4:2:0 is certainly a more aggressive compression scheme than the previous examples, but that doesn't mean it's bad. In fact, it's far more common than the other two. Some of the most common codecs, like H.264, use 4:2:0 subsampling as part of their specifications. Even some high-quality delivery formats, like some Ultra HD Blu-Rays, still use 4:2:0.

## Chroma Subsampling in Your Workflow

Your first thought might be that more color is always better. While this is true in terms of absolute visual quality, it might be false for other considerations, like cost and performance. For many applications, 4:4:4 color is simply beyond the scope of technical necessity,

and for some workflows it's purely overkill. For example, most web video platforms (like YouTube) encode all content to 4:2:0. So, if you spent a lot of time and resources shooting, editing, and delivering in a 4:4:4 codec, your work will look almost the same as someone who worked entirely in 4:2:0.

However, where high levels of chroma subsampling become an issue is when you want to do large amounts of VFX work or chroma keying. If an image loses too much color information, it can make these tasks really difficult, resulting in far worse quality. After all, your software can't just invent extra chroma information to pull a clean key from a green screen. What's in the image is all there is. So, if your workflow includes lots of compositing and color work, less chroma subsampling is definitely worth the data requirements. Just like with the other considerations for choosing a codec, the technical requirements of your workflow should be the deciding factor.

## Codec Compression Techniques

Codecs use a variety of advanced computational techniques to compress video data, but there are two primary compression techniques that are important to understand; *spatial compression* and *temporal compression*.

Spatial compression is a technique for saving space within individual frames of video. This method is often called *intraframe* compression, and it works by grouping pixels together and storing them as a single color value, called a block. Blocks take up much less digital storage space than all the pixels would individually, especially when large portions of

the image are a similar color already. However, there will be some inherent quality loss for complex images with lots of detail. To overcome this issue, many codecs employ flexible blocks of almost any shape and size, which results in less perceptible loss of quality.

Temporal compression, or *interframe* compression, is a technique for saving space across multiple video frames. It uses the same block compression techniques as spatial compression, but not for every frame. Instead, temporal compression only saves the *difference* between sequential frames. If half of an image is the same between two frames, then this technique only saves the new pixels in the second frame, and just fills in the rest with the unchanged pixels of the first frame. Codecs that use this method are called *long-GOP* (group of pictures) codecs, and they are very efficient at storing information, because they don't need to store the entirety of every frame. However, because every frame does not contain a complete image, the computational cost is higher if you are working with the footage non-sequentially, like when you are editing. In general, codecs that use temporal compression are great for storing information in a small amount of space, but not so great for working with the footage.

Codec compression

Video compression is a wonderful world of mathematics, computer science, and arcane mystery. The good news is you don't need to be a scholar, engineer, or wizard to understand the basics. In this subsection we'll explore video compression so you can make a more informed choice when choosing codecs.

## Spatial compression

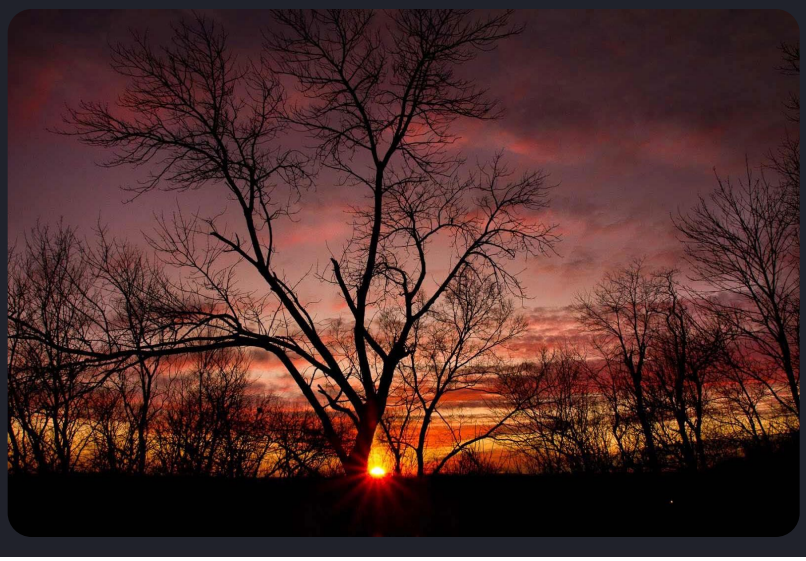
As mentioned earlier, Spatial/Intraframe Compression only involves data within individual frames. To achieve data savings, this method of compression doesn't save the chroma data of every pixel. Rather, pixels are grouped together into groups, called Macro-Blocks, and then an average chroma value is assigned to them.

The exact shape of macro-blocks differ between codecs. Some codecs only use macro-blocks of a fixed size, while others allow some flexibility up to a certain size. In their simplest form, macro-blocks are square or rectangular grids of pixels, like the example above.

However, more advanced codecs can sample blocks of almost any shape or size. In the above image, we can see how HEVC utilizes highly flexible blocks. This provides a much cleaner image than more rigid forms of spatial compression.

Where macro-blocking causes the most quality loss is in images with fine color gradients.

In the above example, notice how the sky fades over a subtle gradient of contrasting colors. This version of the image is very lightly compressed, so the gradient is smooth.



But poor-quality spatial compression can make the gradient choppy and irregular, resulting in banding between colors. This is a direct side effect of throwing away color data, where a large number of differently colored pixels are instead all assigned the same color.

However, despite this issue, intraframe compression is a popular method for a number of different codecs and applications. Popular codecs like ProRes, DNxHD, and some ALL-I (all-Intraframe) capture codecs found on newer prosumer cameras all use intraframe compression.

## Temporal compression

The other category of compression we are concerned with is Temporal/Interframe compression. This method employs the same sort of compression techniques as intraframe compression, but across multiple frames, rather than just one. That means that every frame doesn't contain 100% of the information used to display it. This might seem confusing, but it's actually quite simple.

In the above shot, the subject is seated in front of a static background. Obviously, as the interview progresses, there will be motion on screen. The subject's face will change as they speak, their hands will move, and they will wiggle a bit in their seat. But as a whole, most of the image (the background) will remain largely unchanged from frame to frame. Even a good portion of the subject's body will stay the same in short bursts. So, in reality, a lot of the information in every frame might be redundant if they were all stored equally.

So interframe codecs eliminate this redundant data by only saving changes present in every frame. In this case, that's the subject's face, and whatever hand gestures or body movement they make. How exactly does this work?

This complex process begins after the interframe codec breaks up an image into macroblocks. At this point, an intraframe codec would just encode the image, but interframe codecs are different. instead of directly encoding the values of every block, the compression algorithm searches for similar blocks in adjacent frames. If it finds one, it then references that block in the new frame.

And this works even if the blocks between frames are slightly different. If the referenced block is in a new

position (because of camera movement) it can be placed in the new frame in a new position, by use of a motion vector. This vector shows where the reference block should appear in the new frame, all without creating new blocks.

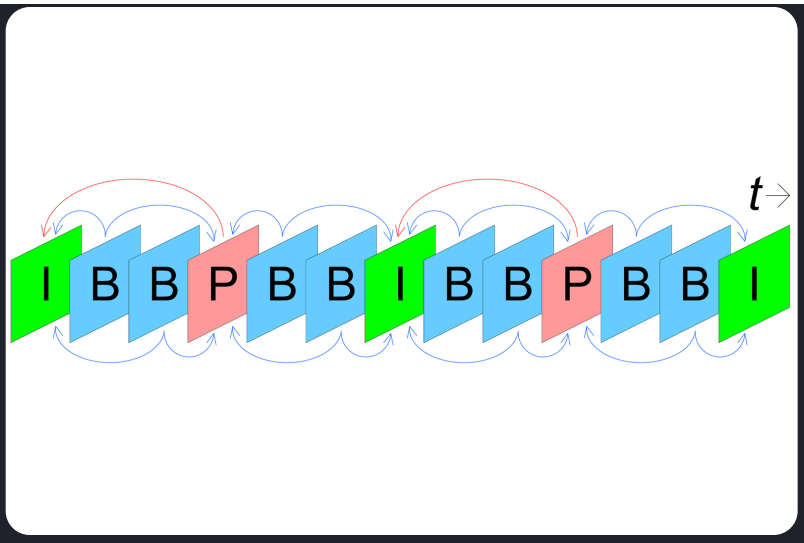
Now, it wouldn't make sense for the codec to use every frame in a clip for this sort of compression. After all, scenes changes have practically no details in common, so it would be fruitless to look for similar blocks in those situations. This is why temporal compression groups similar frames together for more efficient encoding and decoding.

Thus, interframe codecs are called Group of Pictures (long-GOP) codecs. These groups are made up of three different types of frames – I-Frames, P-Frames, and B-Frames.

I-Frames are Intraframe-Coded images. That means that these frames only receive spatial compression. They serve as the reference frames for the rest of the interframe sampling.

P-Frames are Predictive-Coded images. These are frames that rely on a reference frame that comes from earlier in the sequence. Any blocks that are the same between a P-frame and an I-Frame will only be stored in the I-Frame and just loaded into the P-Frame for display.

B-Frames are Bidirectionally-Predictive-Coded images. These are similar to P-Frames, but can use data from images earlier or later in the sequence. The algorithm can search for blocks in either I or P frames to fill in the gaps.




Depending on the compression settings, the number of frames in a group will vary. In general, the more an image stays the same, the more frames can be put in a group without visible artifacts. There can be lots of P and B frames between each I-Frame, or only two or three. It is also sometimes possible for users to set how many frames these groups can contain, which gives more technical control over the image.

Lots of codecs use interframe compression, such as H.264, MPEG-4, AVCHD, and XDCAM. Interframe compression is appealing for use in capture codecs, because they allow you to store more footage on a given storage device. That allows you to shoot longer on set or in the field without changing drives.

## Bit Rate

Finally, you need to understand *bit rate*. Bit rate measures how much data a codec uses over time, usually expressed as *megabits per second* (Mbps or Mb/s). A codec's bit rate determines the visual quality of your video more than almost

any other factor–more than resolution, more than your favorite camera and lens combo, and more than your meticulous color work. Why? Because a codec can only use as much data as the bit rate allows it to use. If a codec has a bit rate of 10 Mbps, then 2 minutes of footage can only use 1200Mb of storage (10Mbps x 120 seconds), period. It doesn't matter if your shooting 720p or 8K, the file size stays the same. Since 8K footage has 36 times more pixels than 720p, the compression will have to be at least 36 times more severe if you try to fit it in the same bit rate as 720p. Yeah, the 8K file might technically still be higher resolution, but it will look awful. When choosing codecs, it's critical that you test the bit rate options to make sure they can reproduce the level of visual quality you need for your project.

[Note: A megabit (Mb) is not the same as a megabyte (MB). 1 byte is made up of 8 bits. In the previous example, 1200Mb equals 150MB (1200Mb / 8bits = 150MB). The 1:8 conversion of bytes to bits is the same regardless of the binary prefix.]

Bit rate

## Calculating Bit Rate

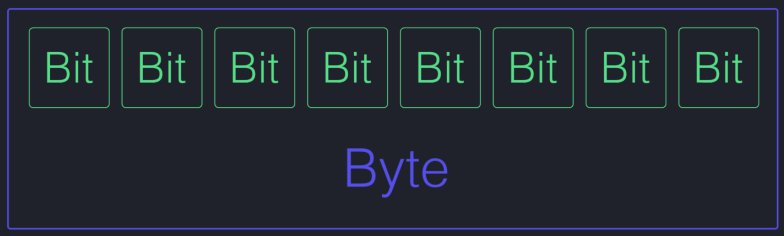
A codec's bit rate can seriously affect the speed and efficiency of your workflow. Using the wrong bit rate at the wrong time can lead to inconveniences, like slow editing, or possibly disaster, like the complete exhaustion of your available storage. To avoid these issues, you really should learn how to calculate bit rate.

The basic formula for bit rate is *bit rate x length of video = file size*. This is a very simple formula that anyone can use. Keep in mind, however, that each of these variables

can be filled in with slightly different units. For example, bit rate can be expressed in kilobits per second (Kbps or Kb/s) or megabytes per seconds (MBps or MB/s). A video's length can be in seconds, minutes, and hours. And a file size can be a few kilobytes, or up to a couple terabytes (pray that it's not).

To get a handle on these different units, let's first answer a question. What is a bit?

A *bit* is just a binary unit of information. Computers store bits as a 0 or a 1, and combine them together to store increasingly complex information. When 8 bits are put together, they form a *byte*. Converting between the two units is very easy. To convert bits to bytes simply divide the number of bits by 8. When converting the other way, from bytes to bits, just multiply the number of bytes by 8. Simple.




Well a byte really isn't that much data (anymore). So, we use prefixes to quickly quantify how many bits/bytes we're talking about:

One kilobit/byte is one thousand (1,000) bits/bytes

One megabit/byte is one million (1,000,000) bits/bytes.

One gigabit/byte is one billion (1,000,000,000) bits/bytes.

One terabit/byte is one trillion (1,000,000,000,000)

bits/bytes.

Notice that every step up in prefix increases the amount of information by 1000. So, 1 gigabyte is 1000 times larger than 1 megabyte. 8 billion 1s and 0s (one gigabyte) certainly sounds like a lot, but that's the scale it takes to store, process, and display most HD video files.

Here's an example. Imagine you have 2 hours of footage you need to archive, so you're transcoding it to DNxHD 175. How much storage will you need?

Using the formula above, let's start with bit rate. The bit rate of DNxHD 175 is 175 Mbps (hey, isn't that handy?). Since we want to know how much storage is necessary, we should convert the bit rate to *bytes per second* (because file size is almost always expressed with bytes rather than bits). Simply divide the bit rate by 8, and you're there.

175Mbps / 8 bits = 21.875MBps

Now, let's factor in the length of the footage. We could figure out how many seconds are in two hours of footage, but it's easier to just convert our bit rate to a per minute value. So, since our bit rate is currently measured by the second, we need to multiply it by the number of seconds in a minute. That's 60.

21.875MBps x 60 secs = 1312.5MBpm (megabytes per minute)

Now, the final step, multiply our new per-minute bit rate by the number of minutes in two hours. That's 120.

1312.5MBpm x 120 mins = 157500MB.

Now we know exactly how much storage we will need to transcode into our archival codec. But we can make this answer a bit easier to read if we convert it to gigabytes (the most common unit of video file sizes). Using our table above, we know there are 1000 megabytes per gigabyte, so all we need to do is divide our answer by 1000.

157500MB / 1000MB = 157.5GB

See how easy that was?

## Adjusting Bit Rate

Now that you know how to calculate bit rate, you can easily figure out which codecs fit within your workflow resources. But once you've narrowed the list, how do you know which codec (and corresponding bit rate) are the best options for your project?

Well first you should know that bit rate is adjustable. While many codecs have a default bit rate on certain devices/pieces of software, you will usually have some control over this aspect of the codec. Some codecs (like DNxHD and ProRes) have bit rate presets (see DNxHD 175 in the example above) that are optimized to work for particular resolutions and framerates. This takes a lot of the guesswork out of choosing a bit rate for the codec, because the different presets give similar quality for different formats of video. Other codecs (like H.264) allow you to variably select almost any bit rate. This gives you more precise control over the size of your footage, which can be useful when storage or bandwidth is limited, but can take more time to find the exact settings that preserve a desired level of detail.

Generally speaking, you want to use the highest bit rate that your workflow resources can tolerate. A higher bit rate will allow a codec to produce higher quality images. However, this is not necessarily true when comparing codecs against each other. Different codecs vary widely on how they use compression techniques, so choosing codecs based on bit rate alone is not a good idea. For example, a newer codec like H.264 at 10Mbps will produce much higher quality images than an older codec like MPEG-2 at 20Mbps. H.264 is just that much more efficient than MPEG-2, so it can fit much more detail into less data. Only compare a codec against itself when considering bit rate alone. If you do that, then yes, a higher bit rate will produce a higher quality image. But only to a point.

There is a limit to how much increasing bit rate will improve an image. Most codecs have a "sweet spot," or range of bit rates where image quality and file size are optimized. For example, H.264 at 1080p is widely believed to have a sweet spot between 15 and 25 Mbps. Using a bit rate below the sweet spot can degrade the visual quality of footage, and going above it will use more storage only a minor increase in quality. Sure, you do get increased detail with H.264 if you jump from 20mbps to 200mbps, but it will not be as discernable as the jump from 10-20mbps. Plus, your storage requirements will skyrocket. In most cases, it's not worth the added computational/logistical costs to increase bit rate beyond a certain point.

In short, there is no ideal bitrate. Your best option will depend on the image you are recording/working with, its resolution, how you want to manipulate the image, and how much data your workflow can handle. For some shots, low bit rates will look fine, especially if there isn't much detail in the image to preserve (think of an

interview with a still background). For other shots, like action sports and chase scenes, where almost everything in frame is changing constantly, you will need a much higher bit rate to achieve an acceptable quality. Experiment with different bit rates with codecs you already use, and see how they impact the image quality. As with everything that surrounds codecs, you need to thoroughly test your bit rate before the project begins.

## Types of Codecs

As we established already, there is no single codec that is good for every part of the workflow. Simply put, the codec you use in-camera will not be the same codec that makes it to a viewer's screen. The characteristics and specifications you need for these two applications (and every step of post-production in between) are radically different. If you tried to use the same codec for every step of post-production, you would have to sacrifice something along the way. So, you will need to pick codecs that make each part of your workflow the easiest, while also maintaining quality.

Your codec journey begins in the camera, with the codec you use to capture your footage. The best capture codecs have deep bit depths, low chroma subsampling, high bit rates, and do not use long-GOP compression. There is a huge variety of capture codecs, but if it meets these criteria then it's likely a viable option. As a rule, you will want to use the best codec you can for capture, because it will be the source for all other codecs you transcode to down the line. You can always convert high quality footage to lower quality files, but it doesn't work the other way around.

Once your footage is offloaded from your camera, your next

codec choice will be the codec you use to edit your footage. Editing requires speed and ease-of-use above image quality, so it's a good idea to transcode your original camera files to a codec that generates much small files. However, be sure to avoid long-GOP codecs (like most versions of H.264) for editing. These codecs do make small files, but your computer will have to work much harder to keep up with all the referenced information as you scrub around the timeline. That may slow down your editing significantly, so make sure your editing codec only uses spatial compression (like DNxHD or ProRes). Also, don't be embarrassed if your computer needs to user a lower-quality codec for editing. It was only a few years ago that even big-budget Hollywood movies were being edited at 480p. It's just that much faster and easier. Besides, once you've locked down the edit, you will be relinking to the higher quality files anyway that will replace the low-quality editing codec.

After the edit is complete, you will need a codec for color correction and VFX. Often times you can just use the original camera files that were conformed into your video timeline, which ensures you are getting the best quality image possible. But if you decide to use a separate codec for these processes, you will want one that is preserves even more information than the capture codec.

When the time comes to export your video for delivery, you might think that a bigger codec is always better. In reality, though, the delivery medium always dictates the codec. We cover those concepts in more detail in the delivery section.

Once you deliver the final version of your video to its intended destination(s), you will need to export in a different codec for archiving. As happens in this industry, the software and hardware we use changes frequently, and clients sometimes ask for a higher resolution or higher quality files years after delivery. In those cases, opening the original project files and re-exporting is not always possible. So, you need a good

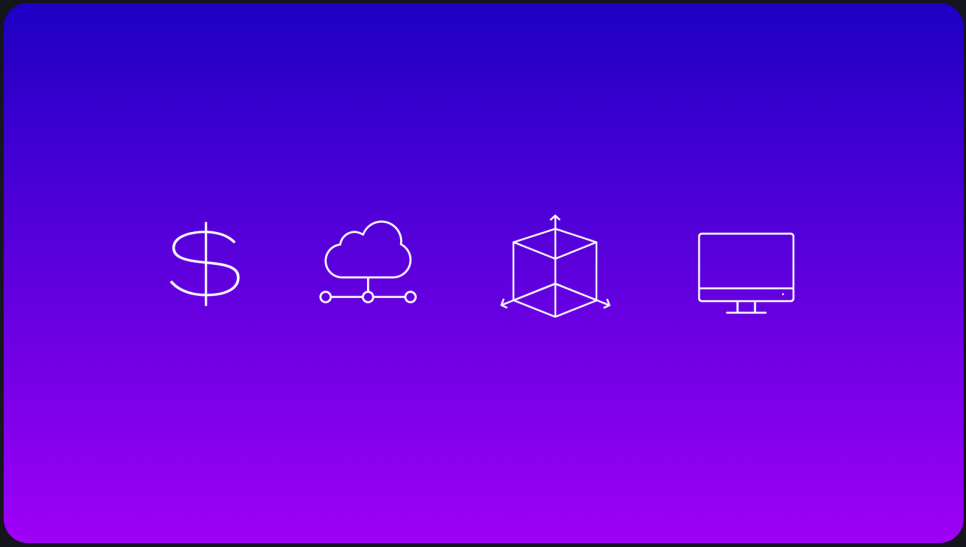
codec for your archived files. It should be extremely high quality, with as high a bit rate as possible, so that you will get the best possible results when you need to transcode it in the future. It's usually worth the tradeoff of storing a very large file.

There are hundreds of codecs out there, and at times it can seem overwhelming. You might wonder how you're even supposed to find the best options for your project. However, there is another factor to consider when picking codecs that should make choosing them easier; their popularity. Popular codecs are more likely to work on your computer, on your client's computer, and on both of your new/upgraded computers five years from now. Also, popularity breeds support. If something goes wrong or you have a question, it's easier to find help if thousands of other users are posting about it online. When you're picking codecs for every step of your project, you can save a lot of time (and heartache) by using codecs that are popular.

If you are ready to pick codecs for your project, you should check out the Frame.io Codec Comparison. This tool will give you a good idea of how the most popular codecs handle your footage, and help you decide what works best for your workflow.

Generally speaking, you should aim for the highest-quality codec that your camera (or your budget) can capture. When we say "highest quality", we mean that you want to capture as much information as possible–so you want less compression, higher bit rate, higher bit depth, and less chroma subsampling (click on the box above for more explanations). The more information you have when you capture, the more flexibility you will have later, especially in color-correction and VFX.

[Check out the Codec Comparison to see for yourself how different codecs impact image quality.]



## Factor 1: Cost

The first consideration for many is cost. Generally speaking, the more expensive the camera, the better the available codecs are, but there are some exceptions. There are some "sweet spot" cameras that offer excellent codecs at a reasonable price. Panasonic's GH series (especially in the early days when the GH2 was hacked) are known for offering better codecs than the other cameras in their price range. (Panasonic's GH5 can record internal 4:2:2 10-bit recording at 4K for under US$2,000).

Blackmagic Design (along with the Magic Lantern hack for Canon DSLRs) brought raw recording options to lower-budget cameras a few years ago. These cameras are the exceptions, however.

## Factor 2: Storage

The second factor to consider is storage space. High-quality codecs tend to be higher bit-rate, which means that the files are larger. You need to be prepared to store and back up all of those files as you're shooting. You may also have to upgrade your memory cards in order to be able to record the high-bitrate data.

If you're shooting solo, you may choose a lower-quality codec because it allows you to change memory cards less often and focus on the story instead.

## Factor 3: Finishing

Another factor to consider is how much color-correction and VFX (collectively referred to as finishing) you plan to do. If you're going to be doing very minimal color-correction and no VFX, then you can probably get away with lower bit-depth, chroma subsampling, and macro blocking that come with lower-quality capture codecs.

EFILM Senior Colorist Mitch Paulson in finishing a session.

It's mainly in color-correction and VFX that you'll see the benefits of higher-quality codecs.

## Factor 4: Editing Hardware

The last factor to consider is your editing hardware, because most capture codecs are not well suited to editing without a high-performance computer. H.264 and some raw files formats require a powerful CPU and/or GPU to edit smoothly. Very-high-bitrate codecs may require high-speed hard drives or shared storage. Unless you happen to be shooting with an edit-friendly codec, you may have to transcode your files to another codec for a smooth editing process, and that can take time. For most people, transcoding the footage isn't a huge issue because it can be done overnight or on a spare computer.

If you're working with very tight turn-around times, however, you may choose a codec that will allow you to start editing *immediately* after a shoot, even if that means a higher cost or a sacrifice in image quality.

# Log Recording

Once you've chosen your codec, you're also going to need to decide whether to record in log or not.

Recording your video in log has the significant benefit of allowing you to retain more color information than "standard" (non-log) video, especially in the highlights and the shadows.

Colorists love log footage because it gives them a lot more room to make changes in the color grade. If you record log footage with a high-quality codec (the codec you choose is important), you will have many of the benefits of working with raw footage (e.g. greater latitude and flexibility in color correction; greater retention of details in highs and lows; etc.), but without the requirement of extra processing power.

Nearly every camera maker has their own flavor of log, but they all do essentially the same thing.

Log vs Gamma-corrected Video

To understand log video, we first need to understand how video information is captured by your camera.

## Cameras vs Eyes

As light enters the lens and hits the sensor, every photosite (pixel) measures the discrete amount of light it receives. These measurements are linear, where a doubling of light intensity (number of photons) equals a doubling in signal output. Each pixel's signal is then sent from the sensor to an image

processor, where it is encoded into digital values (more on that in a moment).

But this isn't how humans perceive light. Our eyes don't detect light in terms of absolute, discrete measurements. Instead, we perceive brightness in a relative, nonlinear fashion. In fact, human vision is approximately logarithmic in how it perceives light. What exactly does that mean?

To illustrate, imagine you are in a completely dark room. Obviously, since there is no light, you cannot see anything. Now imagine you light a candle. This will let you see a little bit, and if you had a camera with you in the room, it probably could as well.

But let say you want more light, so you add a second candle. Physically speaking, this will double the number of photons being emitted in the room, and a camera's sensor will actually detect and measure those photons as being double from before. Your eyes, however, will only perceive that there is a relative doubling in brightness (i.e. a movement from one step to a step twice as bright).

Again, imagine you light another candle. This is where our eyes and camera diverge. Three candles emit three times the number of photons as a single candle. This is what your camera's image sensor detects, so it has three times the brightness information as before. But your eyes do not. Rather than being three times as bright as a single candle, it's only 2.5 times as bright. But why? Because the movement from two candles (the second step) to three candles (the third step) is only half as big in relative terms.

By now you can probably see the difference between linear and logarithmic perception of light. This is the difference between how our eyes and cameras see. To continue the above example, a fourth candle would be four times the

amount of light of a single candle from the camera's perspective, but would only be three times as bright to our eyes. Eight candles would be eight times the number of photons hitting the image sensor, but only four times as bright as perceived by our eyes.

Adding a 17th candle would barely make a difference to our eyes' perception of relative brightness, but the camera would continue to measure a linear increase in the number of photons. And if our eyes barely notice any change in brightness between 16 and 17 candles, imagine how much smaller the perceivable difference is between 99 and 100 candles. You'd barely notice any difference in brightness at all.

So what has all this got to do with log footage?

## Gamma-Corrected Video

The disparity between our eyes and cameras means that not all light information is created equal. We simply don't perceive variations in the brightest parts of an image as much as we do variations in the darkest parts. Knowing that, why should a camera allocate the same amount of space to all light information equally? In reality, it doesn't.

Since storage space is limited, engineers took advantage of the situation. They developed a method that preserves image quality across the parts of the image our eyes are most sensitive to (like shadows), but without increasing bandwidth or storage requirements.

This process is called gamma encoding, and it takes place inside our cameras. After the analog light signals are measured by the image sensor, that information is sent to an image processor which converts it to a series of digital values.

During this conversion, a mathematical function translates the camera's linear light sensitivity to a more logarithmic scale. This is gamma encoding, and it has the effect of stretching the darker tonal ranges of an image across more bits of information, while compressing the brighter tonal ranges into fewer bits.

This is great, because it allocates digital information more efficiently across tonal ranges (in a way that is similar to how our eyes perceive them). More bits are used for tonal ranges our eyes are sensitive to, and fewer bits are given to the tonal ranges we perceive less.

However, a gamma-encoded image won't look the way the actual scene did to our eyes. If we want the image to look the way we saw it in the real world, we need to restore its linear brightness.

This process is called gamma correction, and it takes place in our computers and displays. Basically, gamma correction is just the opposite of gamma encoding. This process applies a mathematical transform function in reverse, returning the image to a linear brightness scale.

This is also great, because it represents tonal ranges in an image as our eyes expect to see them.

In short, gamma encoding saves space by throwing away some bright parts of an image our eyes won't really notice, and then uses that available space to store the dark parts we will notice. Gamma correction then displays that preserved tonal information in an image that is more representative of what the scene actually looked to the human eye.

Log Video

This is where log comes into play. Log image profiles apply a mathematical function to the output of an image sensor, before the information is encoded by the image processor. If this sounds like gamma encoding to you, then you're paying attention. The difference is that the mathematical function for log encoding is a logarithmic curve (hence the name). This is a much more aggressive curve, which stretches out brightness information over a much wider set of tonal ranges. That means that the image is much flatter than standard gamma encoding.

This approach uses far more digital information to store the shadows and midtones of an image, which preserves the lower stops with much greater detail. This enables greater precision for manipulation in post, but doesn't increase the storage requirements above those of standard gamma-encoded video. Log provides more subtle steps in brightness (shades of gray) on screen, so you have much more flexibility to adjust the look of an image to be just the way you want.

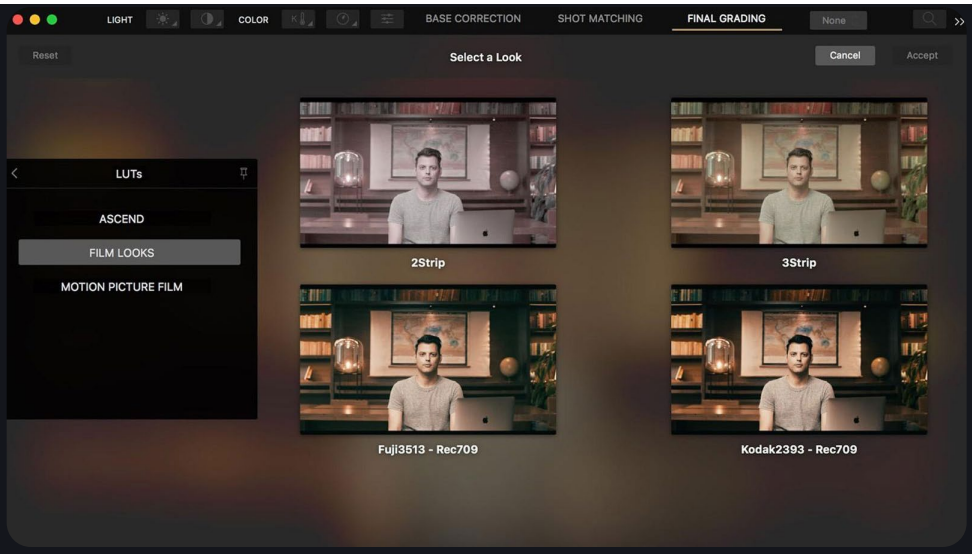
Of course, this also means that when you view the image on most displays it will not look the way it did in the real world. Why is that? Because the standard gamma correction your monitor uses isn't logarithmic, so the image will still look washed out. However, this is easily fixed by applying the corresponding logarithmic function in reverse, which is exactly what some LUTs do. This is basically gamma correction for log video, but you have the ability to tweak it in post.

To summarize, shooting in log preserves more detail across the dynamic range of an image. This preserved detail gives you more creative control over an image, similar to raw footage, but without the enormous file sizes. That means you can make finer adjustments in post-production, but with the same hardware as standard video files. That said, log isn't meant for final delivery/viewing by itself, so you will need to add steps in your workflow to correct the image (using a LUT

or otherwise).

The most obvious downside of recording log is that the image that comes out of the camera looks extremely low-contrast and desaturated. This makes it very hard to judge the images, which can be a pain, especially if you have a client peering over your shoulder at the monitor. Most people aren't used to viewing log images, and so the images won't look right to them.

In order to preview your footage accurately, you will need to add a lot of contrast and saturation. One common way to do that is by applying a Look Up Table, or LUT. A LUT is essentially a color correction preset. You'll choose a particular LUT that's designed to match your particular type of log and make it look like "normal" footage. If you've used Instagram filters, a LUT is the same concept, but much more precise.




## LUTs

A Lookup Table basically works like a digital camera filter – meaning that it is a primary adjustment. It affects the entire image–a LUT cannot reflect keyers, power windows or any kind of tool that isolates the adjustment to a part of the image.

The effect is similar to just adding a blue piece of glass in front of the camera when shooting: The glass turns the image blue and the effect is definite – meaning that if you want less of the blue effect, there is no way of making the glass "less blue". It's similar in your software of choice – it offers you for instance a slider that lets you dial in a blue tint dynamically – however, a LUT you can either load, or not – you cannot load a LUT only half way. If you want to mitigate the effect of the LUT, you need to dial against it with another tool in your software – for instance, load the LUT onto a layer (or node) and dial down that layer's/node's opacity.

That however makes it a bit of a double back and forth: Add 50, subtract 10 – not an ideal process. Therein lies a big advantage of a CDL – but we'll get to that in just a bit.

There are usually two kinds of applications for LUTs: One is technical LUTs, where the LUT is supposed to do a technical conversion from one color space to another, without adding any kind of creative look into the image. Such a conversion can be for instance RED Wide Gamut to Rec709.

Another example is the use of calibration LUTs, where the only purpose is to calibrate the corresponding display in order to show reference colors and light levels.

The other application for LUTs is obviously the creative field. There are lots of LUT-packages out there (as anyone can create them easily) which define certain looks – cold, warm, vintage and whatever you want to call them. Creative LUTs can serve as a starting point for a creative grade. Swapping out a LUT against another one is as easy as changing a glass filter in front of a camera lens, hence it is perfectly suited for quickly shuttling through a number of different looks for a given scene.

Another advantage of LUTs is that they can be loaded into a multitude of hardware devices, like Monitors and LUT-boxes, that can process it on the fly without further delay.

Creative LUTs are a good way to transfer the intended look of a given scene from set to post: During the shoot, the live SDI signal from the camera is being routed through a LUT-box, which modifies the live SDI signal on the fly – being controlled by any software that supports the LUT-box – most established is Pomfort's LiveGrade in this field. The look dialed in for a particular scene can then be saved as a LUT and forwarded to post as a starting point for the final grade. This LUT can also be used to create the offline dailies, that are going to be used in editorial: This way there is already a rough look applied during the edit.

An important thing to understand about using creative LUTs

in post is that they don't free you from actually grading and balancing your clips. Let's say you have two clips of the same scene – one shot in bright sunny daylight, and when the second one got shot, there was a cloud in front of the sun and the whole image got much darker.

If we now apply the same LUT to both clips, the effect will be the same (lets go with the blue-ish tint here again) – but still one clip will show darker than the other, so balancing the two, so they match each other, is inevitable.

Another thing to keep in mind is that low-resolution LUTs can indeed cause banding and artifacts in your image, so don't trust just any free LUT-package out there that comes across.

## LUTs exist in two flavors: 1D and 3D.

1D LUTs are obviously the simpler ones, since they are basically just a gamma curve (and can easily be converted to one): An input value maps to an output value and that's about it. 1D LUTs are mainly used for log conversions (e.g. ARRI LogC to 2.4 gamma).

The most common file extension is .lut – sometimes (but rarely) you can also find them in an XML format. The accuracy of a LUT is expressed in two parameters: Size and Depth. Size defines the number of input values that can be mapped – the more values between 0 and 1 (black and white), the more accurate the LUT is.

And Depth is the same accuracy, but on the output side. If you open up a 1D LUT with a text editor, size is basically the number of entries in there, and depth the accuracy of those entries: With a depth of 8 bit you will find values between 0 and 254, whereas with a depth of 10 bit you will find values

between 0 and 1023 – a much finer grid.

A 1D LUT can contain just one table that maps R,G and B channels equally, or it can contain 3 tables to map each one and even a fourth channel for alpha (if supported by the software).

Now for the 3D LUTs – no, it has nothing to do with stereoscopic 3D.

Instead, imagine a 3D cube, where hue, saturation and lightness are mapped to each axis: X, Y and Z. The same input/output game that we know from the 1D LUTs plays here in 3D space. Effectively, this means that you can e.g. map a red-color pixel directly towards green, whereas with a 1D LUT you can only increase, or decrease a red value. Same as with a 1D LUT, the accuracy of a 3D LUT is expressed through size (of the cube) and bit depth (of the entries inside).

The most common file extensions you will find are .3dl and .cube – but there are more, less common ones out there, too.

An important thing to keep in mind: Not all software applications and hardware boxes support 3D LUTs equally – make sure you have your LUTs in the correct file format, size, and depth for the software/hardware you intend to use them with.

So, recording in log still adds some complexity to the workflow. You have to find the right LUT for your footage, and you also need to manage your LUTs throughout the whole post-production process.

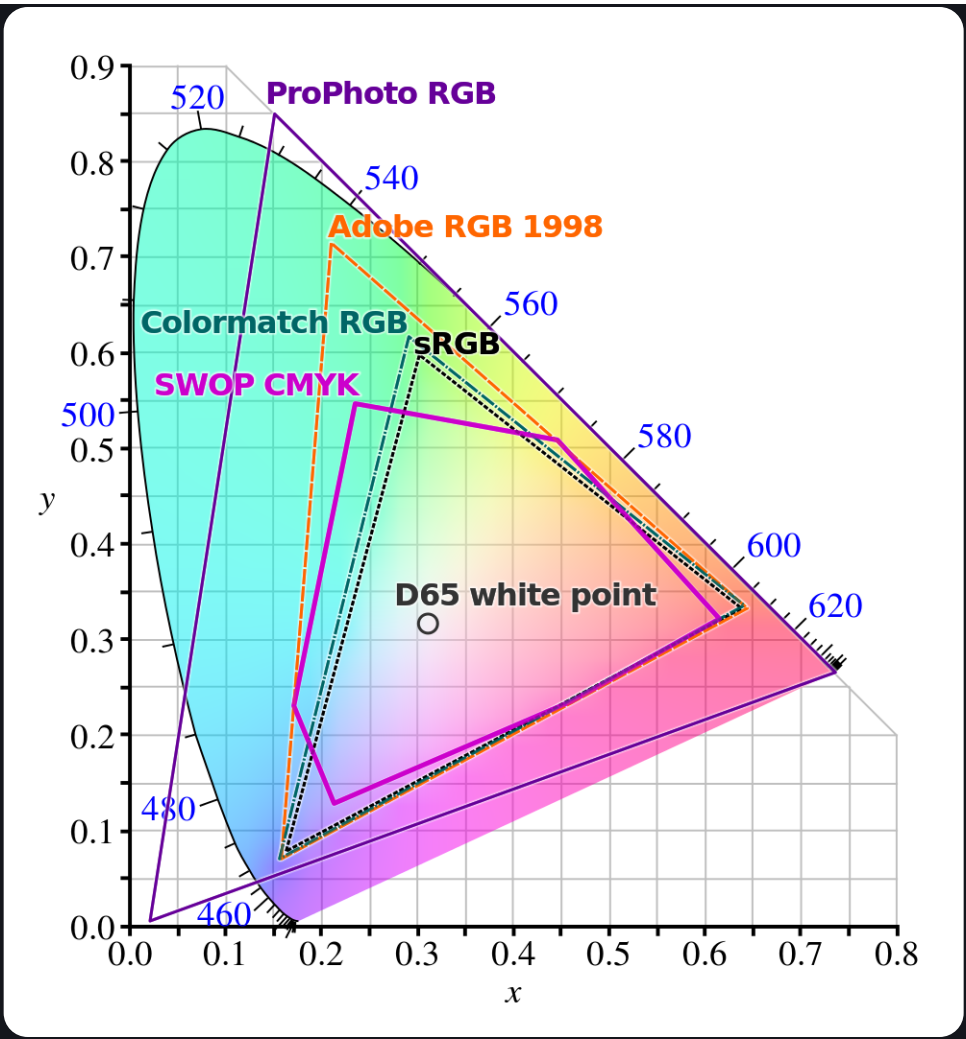
In spite of that annoyance, the image-quality benefits of log recording are significant, and it's now very common. We will assume that you're recording in log for the rest of this guide. If you're not, don't worry. Your workflow will just be slightly simpler because you

won't need to add any preview LUTs.

## Color Spaces

In discussing log images, we've just touched the topic of *color spaces*, which is a fundamental technical concept that underlies all image data and processing.

A color space is a way to describe colors using electronic data. Color is really an attribute of light waves, and light is analog, which means there is an infinite number of subtle shades of color. You can't capture that infinite range of colors in a digital computer, however. Computers need to work with concrete numbers, and so a color space is a method for translating light into numbers.




You can translate light into numbers in many different ways, and so there are many different types of color spaces, some of which have been around for decades, and some of which are brand new. Each of them is well-suited for a particular case, but none of them is best for every situation. Some are good at capturing, and some are good at displaying on monitors, while others are good at displaying on projectors.

Every camera, when it records data, has to record it into some kind of color space (unless it's raw video, in which case you're able to choose your color space later). You can always convert from one color space into another, but you have to be careful, or you may lose some information.

Some color spaces can hold a broader range of colors than others, and so if you convert from a bigger color space to a smaller color

space, your software may have to chop off some of the edges to make it fit. Unless you do it properly, that can have a disastrous effect on your images.

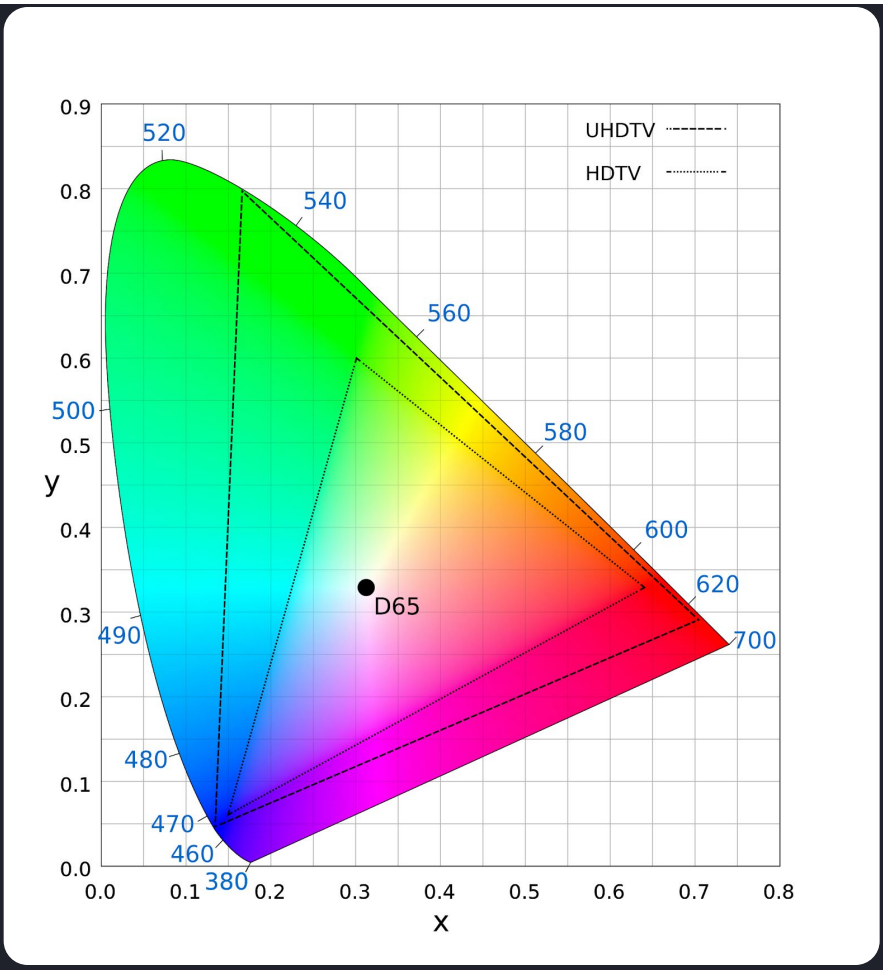
The balance you have to consider when choosing a color space is the same balance that we've discussed when talking about bit depth and codecs. Generally speaking, if you record more information, you will need more storage space, you will need more computer power to process that data, and you will need to spend more time managing and tweaking your workflow to ensure that your color space is preserved.

Thus, high end workflows tend to capture into very large color spaces and preserve them carefully throughout the entire process, especially if the end result may be projected in a theater. Lower-end workflows (especially on projects that will only be viewed on computer screens) tend to go with simpler, smaller color spaces in order to save time and money.

Color Spaces

A color space is a way of organizing colors. In the context of images and monitors, we describe color spaces in terms of their RGB primaries and white point. It is important to know what color space an image is in so that colors can be correctly interpreted and displayed. As part of a workflow, colors may need to be converted from one color space to another, and a knowledge of color spaces can help us avoid loss of information or inaccurate display of colors.

A color space definition consists of the CIE 1931 chromaticity coordinates (x,y) of the three primaries (red, green and blue) and the white point. Plotting a triangle on a CIE 1931 xy diagram provides an outline that helps visualize the visible colors a color space encompasses.




The gamut of a color space is the set of all colors that can accurately be represented in that space. When an image in one color space is converted to another color space, the relative sizes of the gamuts become important as some colors that can be represented in a larger gamut cannot be accurately represented in a smaller gamut. Typically any color space that has a wider gamut than REC709 is referred to as "wide gamut".

Throughout a project, color spaces will need to be chosen. The camera will be recording into a color space (even RAW is a color space in the sense that it encompasses a set of colors the camera can see), images will be in a color space for viewing while being edited, grading will require images to be in a color space and probably transform them all into a deliverable color space.

The process of converting from one color space to another is part of color management and therefore it's important that, along the path the image takes through the workflow, it is not unduly restricted in color space. Once a color space is limited to a smaller gamut, it cannot be expanded back up to a larger gamut – that information is lost. This is analogous to how dynamic range is handled through capture, editing, grading, output whereby the best workflows capture as high a possible dynamic range in the camera (recording RAW to ensure nothing is lost, or using a log recording where, if careful, little dynamic range is lost), only compressing that dynamic range to that which is pleasing on a given display as part of the grading process.

Color Management with Adobe Applications

Contributed by the team at Adobe.

Camera sensor data doesn't have a well-defined color space. Camera sensors are not colorimetric devices, and although the mathematics of color spaces are often used to describe camera data, it is transformed into a color space before use. That color space could be a display space such as Rec. 709, Rec. 2020, DCI-P3, or a more camera-specific color space designed to encompass all the colors a camera captures into a wide gamut space suitable for grading. Examples include REDWideGamutRGB, Arri Wide Gamut, and Sony S-Gamut. These wide gamut spaces will need proper conversion to the display or deliverable color space to ensure accurate portrayal of captured colors. The conversion should be done with care to ensure any colors which are in the camera space gamut, but outside the display gamut are handled in such a way as to avoid clipping which can produce ugly results.

Sometimes a common intermediate color space will be used for all elements of a project. The Academy's ACES project provides a color space for this purpose AP0, but in other

projects the most commonly used source's wide gamut colors space will be used as the common space and the other elements all converted to it.

The display or deliverable color space will be defined by the nature of the project. For cinema release it's typically DCI-P3, for TV and web it's Rec. 709, and for HDR UHD it's Rec. 2020. As a project will usually encompass a range of source material, elements that are not in the target deliverable space will need to be converted. Special care needs to be taken with graphic elements and logos to ensure they display correctly on the final export.

Contributed by Graeme Nattress, Red.

## ACES

The Academy Color Encoding System ("ACES") refers to a format and also a process that goes along with it, which makes it easier to work with deep color spaces in high-end workflows. Working with deep color spaces and high dynamic range (HDR) images can be extremely complicated because there are so many different formats and tools and workflows, many of which are not compatible with each other. ACES aims to solve that problem by creating a single, standardized workflow that can work for everyone who really cares about preserving all of their image data through the entire image pipeline.

If you are producing videos that are only going to end up on the web, then chances are that ACES is unnecessary for you. If you are producing a feature film to be shown in theaters, or if you are investing a lot of money into a project that may live on for years to come (and may be shown on as-yet-unknown technology), then ACES may be well worth your while.

ACES

ACES or the Academy Color Encoding System is an open color management and interchange system developed by the Academy Of Motion Picture Arts & Sciences (AMPAS) and industry partners.

ACES standardizes the color science used in projects of all types and is designed to give the industry a standardized color management system that encompasses production, post-production, presentation, and archiving.

## The Goal Of ACES

In 2004 when the ACES project started, the Academy Of Motion Picture Arts & Sciences (AMPAS) and many industry experts realized a need for an open color management system that was suitable to handle the complexity of modern

digital production.

In an industry with dozens of different camera systems, multiple encoding options, different display devices, etc, the Academy saw a need for a system like ACES to manage color in a precise, yet straightforward way, no matter the camera or display being used.

Many software developers have tackled color management in their own applications, but none of those solutions are industry-wide and open. ACES is an industry-standard color management solution that anyone can use to manage the color pipeline of any project.

Since ACES 1.0 was released in 2014, ACES pipelines have been used on dozens of Hollywood blockbusters, TV series and independent films. The ACES pipeline is available today in many applications including DaVinci Resolve, Filmlight's Baselight, Nucoda, and many others.

## Understanding Capture & Scene Referred Data

One of the most important things to understand about ACES is that its processing pipeline utilizes capture-referred data – i.e. the color science (and sometimes secret sauce) that each camera system uses and bakes into the signal. ACES reverse-engineers that data (through what's known as an ACES Input Transform or IDT) back into the pure linear light information that was in the actual scene in front of the camera. Theoretically, without any camera bias.

This is why ACES is often discussed as being scene-referred or, with the more technical phrase, scene-linear. Abstracting away the camera's bias allows us to get closer to the actual scene that the camera was pointing at.

Additionally, ACES color uses a color space so large it actually encompasses the entire visual locus (everything humans can see) and even light we can't even see! And even the smaller working color spaces are also very large (the particulars of ACES color space(s) will be discussed in a later section).

ACES is a unifying standard – transforming captured referred data from many cameras into scene linear with a common starting point and shared color science.

ACES is also broad – in an ACES workflow you never run into the limitations of working in smaller color spaces. For client handoffs and archival purposes, the ACES color space is so large it is essentially 'future proof'.

The scene linear approach of ACES is only part of the overall pipeline.

The human eye doesn't work in a linear fashion and ultimately projects are viewed on TVs, projectors, etc., that assign gamma or EOTF (Electro Optical Transfer Function) curves that aren't linear. These devices also have much more limited color spaces compared to ACES.

Because of this, ACES also incorporates display-referred color management as part of the overall pipeline. ACES data is parsed through different transforms for different color spaces and display devices.

When it comes to precision, ACES uses Open EXR 16-bit half-float processing which results in 30+ stops of scene referred exposure.

Keep in mind that even though EXR is used, often this processing is just internal to the app you're using and no EXR files are created for you to manage – except in renders.

In sum, these are the benefits of ACES pipelines:

- Camera System Color Science Unification – Because of the capture-referred, scene-linear transform at the start of an ACES processing pipeline (no matter the camera(s) that were used on a project) there is a common color science starting point that all ACES capable applications can use and understand.
- No Guessing For VFX/CGI Workflows – one reason that ACES has been embraced by VFX/CGI heavy films is that they're compositing and working in linear anyway! And that linear data can then be rendered back to whatever is appropriate to the project, or kept linear and given back to a colorist to simply re-apply their existing grade with no look shifting.
- Ready For Wide Gamut/High Dynamic Range – now and in the future. Because ACES is capable of retaining 30+ stops of image data and the gamuts are so large, it's a great match for HDR, wide gamut projects, however those concepts may develop in the future.
- Evergreen Digital Master – One thing that the Academy pushes about ACES is that it allows for a true evergreen digital master because of the ultrawide/high dynamic range nature of the system.

## The 'Parts' Of ACES

Even though ACES and its various transforms are mathematically complex, you can understand ACES better by understanding what each part or transform in the pipeline does.

Here's the terminology for each of these transforms (after we define the transforms, in the next section, you'll see how they work together in an ACES pipeline):

- ACES Input Transform (aka: IDT or Input Device Transform) – The Input Transform takes the capture-referred data of a camera and transforms it into scene linear, ACES color space. Camera manufacturers are responsible for developing IDTs for their cameras but the Academy tests and verifies the IDTs. In future versions of ACES, the Academy may take on more control in the development of IDTs. IDTs, like all ACES transforms, are written using the CTL (Color Transform Language) programming language. It's also possible to utilize different IDTs for a camera system to compensate for different camera settings that might have been used.

- ACES Look Transform ( aka: LMT or Look Modification Transform) – The first part of what's known as the ACES Viewing Transform (the Viewing Transform is a combination of LMT, RRT, & ODT transforms). Provides a way to apply a look to a shot in similar way to using a Look Up Table (LUT). It's important to note that the LMT happens after color grading of ACES data and not every tool supports the use of LMTs.

- RRT (Reference Rendering Transform) – Think of the RRT as the render engine component of ACES. The RRT converts scene referred linear data to an ultrawide display-referred data set. The RRT works in combo with the ODT to create viewable data for displays and projectors. While the Academy publishes the standard RRT, some applications have the ability to use customized RRTs (written with CTL), but many color correction systems do not provide direct access to the RRT.

- ACES Output Transform (also known as the ODT or Output Device Transform) – The final step in the ACES processing pipeline is the ODT. This takes the ultrawide, high dynamic range data from the RRT and transforms it for different devices and color spaces like P3 or Rec 709, 2020, etc. ODTs like IDTs and RRTs are written with CTL

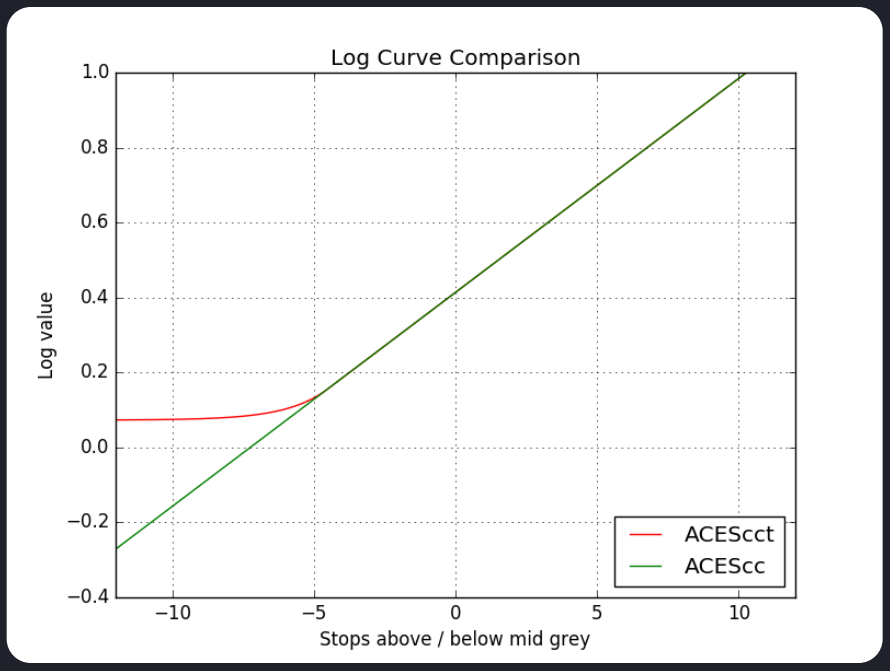
There are also three main subsets of ACES used for finishing workflows called ACEScc, ACEScct and ACEScg:

- ACEScc uses logarithmic color encoding and has the

advantage of making color grading tools feel much more like they do when working in a log space that many colorists prefer.

- ACEScct is just like ACEScc, but adds a 'toe' to the encoding so that when using lift operations the response feels more similar to traditional log film scans. This quasi-logarithmic behavior is described as being more 'milky', or 'foggier'. ACEScct was added with the ACES 1.03 specification and is meant as an alternative to ACEScc based on the feedback of many colorists.

- ACEScg utilizes linear color encoding and is designed for VFX/CGI artists so their tools behave more traditionally.




While ACEScc, ACEScct and ACEScg and the transforms they use are what you'll most often see when it comes to ACES, there are some additional terms you may encounter:

- APD (Academy Printing Density) – AMPAS supplied reference density for calibrating film scanners.

- ADX (Academy Density Exchange) – Used for scanning film and getting those scans into ACES – similar to the Cineon system for scanning.

- ACESproxy – uses logarithmic color encoding and is a integer, range limited version of ACEScc. ACESproxy is meant to be used on set with compatible equipment over SDI.

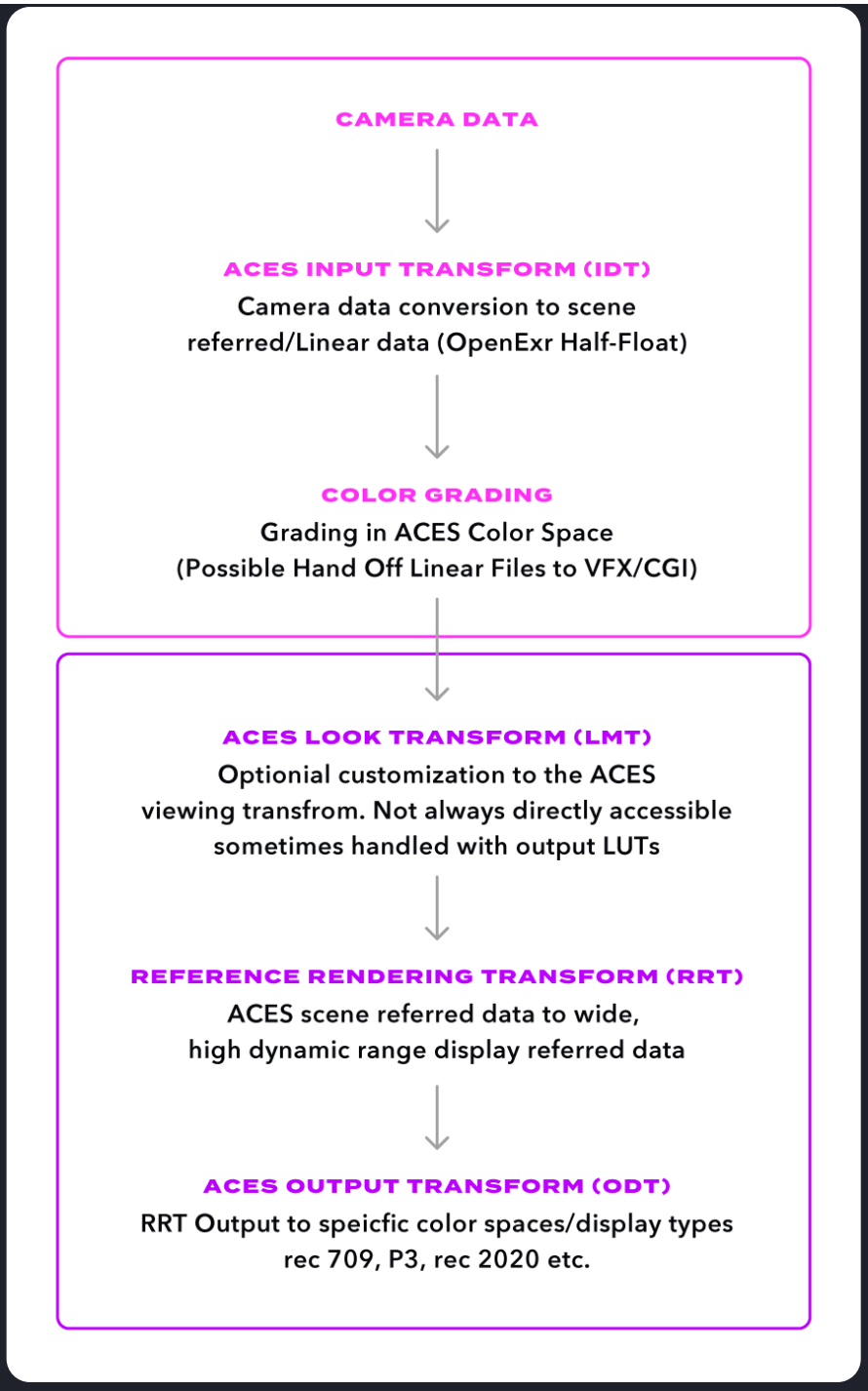
## The ACES Pipeline

Now that we've defined the transforms used for ACES, understanding how the various transforms combine to form an ACES processing pipeline is pretty straightforward:

Camera Data -> Input Transform -> Color Grading -> Look Transform (optional) -> Reference Rendering Transform -> Output Transform

As mentioned, ACES is hybrid color management system of scene referred/scene linear and display referred data.

In the graphic below, the various ACES transforms fit into the scene referred (top section) and display referred (bottom section) part of the pipeline:




## Understanding ACES Color Spaces

When ACES is discussed, you'll often hear some highly technical phrases to describe ACES: 2065-1, AP0, AP1, Rec.2020+.

What do these terms mean?

SMPTE ST 2065-1 is the SMPTE standardization of ACES.

While this standard has many parts, in daily usage ACES 2065 has come to mean the full linear version of ACES that has a larger gamut than the visual locus set of primaries (red, green & blue). ACES 2065 has a set of primaries known as AP0 (ACES Primaries 0).

A standard CIE1931 plot is a good way to compare ACES to other gamuts. The horseshoe of colors represents the human visual locus – all the colors we can see. The plot on the left shows ACES AP0 primaries compared to other gamuts, you can also see it's larger than the entire visual locus. The plot on the right shows ACES AP1 primaries compared to other color gamuts. AP1 primaries are used in ACEScc, ACEScct & ACEScg

2065-1 AP0 is mainly meant for archival and file exchange. In real-world usage – grading, vfx etc, 2065-1 AP0 is not used. Instead, a smaller set of primaries called AP1 (ACES Primaries 1) are used. Also known as Rec2020+, AP1 primaries are slightly larger than Rec. 2020 which is championed as an ultra wide gamut for the future, and it's the AP1 Primaries that are used in ACEScc, ACEScct, and ACEScg.

## ACES Versions & Open Source Development

Currently, the official version of ACES is 1.1 but many applications are not supporting that version yet, although version 1.03 is widely supported. Unless there are major future changes and/or needs for end users as determined by AMPAS, you will continue to see the major release versions in software and probably not the minor .0x releases even though

software manufacturers behind the scenes may implement the .0X releases for performance and bug fixes.

Projects created using older versions of ACES can be opened using a newer version of ACES that an application uses, but this may cause some differences in image representation compared to the original project. Some applications offer the ability to switch the version of ACES being used to an older one to match the ACES version that was used in the original project.

It's also worth noting that ACES development is 'open source'. AMPAS manages and curates the development of the system but if you're mathematically/color science inclined, you can contribute to the development of the system or observe the development process by visiting the GitHUB repository the Academy maintains for ACES:

https://github.com/ampas/aces-dev

Also, as mentioned, ACES transforms are written in CTL or Color Transform Language. The Academy also maintains a GitHub repository for CTL:

https://github.com/ampas/CTL

## ACES Central

If you're less technically-inclined, you can also visit ACES Central. ACES central is a site that AMPAS maintains geared towards discussion of ACES and is frequented by end users and ACES certified partners.

Contributed by the team at Mixing Light.

# Raw Video

Recording raw is becoming much more common with many of the high-end cameras (and some of the low-end ones) providing raw recording.

Raw footage has many of the same benefits of log, but taken another step further. Whereas log footage gives you lots of extra dynamic range compared to standard video, raw footage gives you a bit more.

## Dynamic Range

"Dynamic Range" is a term used to define the range of tonal detail that a camera can capture between its noise floor and clipping. Simply speaking it is describing the ratio between the brightest and darkest tones that a camera can record within a single frame.

Higher dynamic range matters when an image you're shooting has a drastic difference in lighting – for example, if you're inside a location with bright windows and a dim interior. The light coming from the windows is much brighter than the light within the walls, so either the light from the windows will be too bright and overexposed (so they'll just appear white), or the walls will be too dark and underexposed (so they'll just appear black), and you'll end up losing detail. Here are some good example images of this scenario.

A higher amount of dynamic range allows you to shoot these images while retaining as much information as possible.

A camera's dynamic range describes how much subject detail can vary before being recorded as featureless black or white. Having a higher dynamic range therefore improves exposure and post-production flexibility, broadens the possibilities for dramatic and unevenly-lit scenes, and enhances image quality and detail.

Dynamic range is typically described in terms of "stops,"

where each unit increase translates into a doubling of dynamic range. The language of "stops" comes from the days of physical film cameras and is a very common way to talk about amounts of light. Current high-end digital camera sensors can capture over 13 stops of dynamic range making it comparable to color film negatives. At this threshold of dynamic range, captured images meet or exceed the capabilities of existing displays, but with the release of upcoming HDR display technology the gap is getting smaller.

Contributed by the team at Red.

Both log video and raw video come in many different flavors, however, so it's not always easy to compare them directly. It's possible to have very highly-compressed raw footage, which may look worse than log footage with a high-quality codec. If you're looking to get *the absolute best* color that you possibly can out of your sensor, however, your best bet is to shoot raw with low or no compression.

Apple ProRes Raw image from Frame.io's Apple FCP X integration promo film.

Raw footage also has the extra bonus that, if the footage was accidentally recorded with an extreme color cast (much too yellow,

or much too blue), that can be easily corrected. Log footage does not have as much of an ability to correct for color mistakes. Those mistakes are rare, however.

In spite of the image-quality benefits of raw, there are two major downsides: larger file sizes and extra processing requirements.

With a few exceptions, raw video is almost always much larger than non-raw video. That means more memory cards, more hard drives, and more time spent copying files. Depending on the type of raw we're talking about, you could have 5 times as much data as a non-raw codec.

Raw video also takes significantly more processing power in order to view, edit, or transcode. When you're transcoding your dailies (in the next section), you're going to need beefier hardware or else a lot more patience.

RAW Video

## Fitting Raw into the Broader Workflow

In the big picture, raw video represents the highest quality capture of your images. It can be very demanding for your post-production process. Filmmakers use it because it is the most forgiving and flexible format for the color grading process. Thankfully, there are some advanced methods of raw capture that ease the pain of handling raw, and produce an elegant post workflow with stunning results.

What is Raw?

Cameras gather light. Digital cameras interpret that light to make an image. Raw video represents the *uninterpreted* light that your camera sees, recorded as ones and zeros, before it has made an image file.

Raw or RAW?

Raw is not an acronym, and so it doesn't properly need to be capitalized. Several large companies have begun referring to it in all-caps, however, which has confused a lot of people. The two terms mean exactly the same thing. At this point, it's hard to say that either one is wrong, but we prefer it without the all-caps.

## Why Should You Care About Shooting in Raw?

Most of the time, when light hits a camera's sensor, the camera sends that data through a process. That process renders color according to your white balance, assigns tones based on your ISO setting, applies noise reduction, sharpens the image, corrects for lens profiles, and compresses and encodes the data into a video file.

When you shoot raw video, none of that processing is "baked" into the video file. You get to adjust those values while

shooting, or in post-production. In other words, you get to "develop" your digital film with just the look that you want. When shooting raw, you typically end up with more usable dynamic range, more flexible colors, and the opportunity to apply the kind of sharpness and noise reduction that you want, instead of getting stuck with the manufacturer's preferences. Raw video gives you the most artistic freedom to experiment and fine-tune the look of your film. That's why serious filmmakers want to shoot in raw.

## Debayering

The sensor on your camera has tons of little dots called pixels. They gather the light.

Those pixels lie behind little colored filters arranged in a specific pattern. The most common one is the *Bayer* pattern, named after Bryce Bayer who invented it at Kodak. This array of colored filters goes by the name CFA or colored filter array. It looks like little red, green and blue tiles – just like a mosaic painting.

The surprising thing is that there are two green filters for every red and blue one. That pattern mimics the ways our eyes perceive color. That is terrific for capturing beautiful images, but before you can see a usable picture, that image data needs to be *debayered* or *demosaiced*. The camera or the computer simply needs to process all of that info, extra green included, and produce an image file. But processing all that data on the fly can tax your computer hardware. However, with some smart software and powerful hardware you can take advantage of the flexibility and power of raw in post.

## Is Raw the same as Log?

Raw and "log" are not the same thing. Many video cameras give you the option of shooting in a "log" or "flat" color space if you're not recording raw. The image lacks contrast and saturation. But you are recording the same about of image data across the entire spectrum of your exposure. That can make it easier to perform color corrections in post. However, this isn't raw video. The camera has still interpreted the light to make an image file. White balance, noise reduction and sharpening are still baked in, so you don't achieve the same level of flexibility in post.

## How is Raw Recorded?

You can record raw video in some cameras, like those from RED and Blackmagic Design. Other cameras use external modules or monitor/recorders to record raw. There are even some "hacks" that allow for DSLR cameras to record their raw data, and they can produce some nice images. But those hacks aren't necessarily the most stable, so they remain the domain of the bravest and most experimental of souls.

## Practical Advantages of Shooting in Raw

Raw shooters often fall into two categories. The first category contains those looking for maximum image *quality*. The second category leans toward those seeking maximum *flexibility*. If you find yourself shooting a feature with controlled lighting, where you are trying to squeeze the best look out of every scene, you probably want to shoot in raw. Your camera will record the most image data possible resulting in lovely images.

But less obvious are those shooters who operate in changing

lighting conditions. They are greatly benefited by shooting raw because their cameras are operating with the greatest flexibility to adjust for exposure, color temperature, and tint. That flexibility will enable far more options in the color grading stage of post-production. It is often the filmmakers who engage deeply in both production and post-production work that become the most enthusiastic about raw capture. They observe the benefits of quality and flexibility on set and in post. That allows them to make spur-of-the-moment decisions in production knowing that they will have the tools in post to accommodate for dynamic on-set changes in exposure.

## Green Screen and VFX Capture

Visual effects artists get an enormous benefit when the primary footage has been captured in a high-resolution raw format. Many non-raw codecs use chroma subsampling to save space on memory cards, but chroma subsampling can cause lots of issues for VFX work, especially on green screens.

Raw never uses chroma subsampling and thus avoids the potential for nasty jagged edges when keyed. Separating the subject from the green screen becomes a lot easier. Additionally, correcting the "spill" (the amount of green reflected on your subject) gets simpler as well because raw gives you more color data to work with. This is even true if the application doing the keying uses an image downsampled from the raw. So you might shoot in 6K raw, and transcode to 4K ProRes 4444, to do the keying (perhaps because of hardware limitations). The result will still be superior to something shot natively in 4K ProRes 4444.

When it comes to VFX, the higher the original image quality, the easier the workflow.

## Flying Solo and Shooting Raw

Solo shooters are juggling a lot of hats. Audio, exposure, focus, battery life, stabilization, schedule and more. So why in the world would you want to shoot in raw? Because raw can hide a multitude of sins. It gives you an edge when facing changing lighting conditions and can help to compensate in situations where you might have more time in post than you do on set. Some cameras offer compressed versions of raw. In this way, you're still keeping the flexibility in post, but you are getting the file sizes down to something more manageable. They won't be small, like h.264, but the flexibility is awesome when you need it. And the solo shooter can use every advantage that they can get.

## The Impact of Demosaicing on your CPU and GPU

When your raw data gets interpreted into a viewable image it is called demosaicing or debayering. That can be a pretty computationally-heavy task. Depending on the type of raw you're using, your computer may be able to leverage your GPU to help with your CPU's interpretation on this data. Some graphics cards are more efficient than others when demosaicing certain raw codecs. For instance, RED just announced that NVIDIA's new Touring architecture will allow for 8K real time playback at "full debayer." This means that with a high enough model, you'll be able to see 8K raw footage playback in real time, even if you lack a massive CPU. Cards like this won't come cheap initially, but it is amazing how fast the technology is moving. So it is important to consider the entire "image pipeline," from acquisition to post, when choosing a camera and codec for your next production.

## A Hidden Benefit: New Color Science

One awesome advantage of shooting in raw is that old footage can become new. As a manufacturer releases new versions of their "color science" (the method of interpreting and displaying colors), you can go back to old footage and apply the new color science. This actually makes the old footage look better! It's a great thought to realize that stuff you shoot today will actually improve in appearance as time goes on. Colors that were out-of-gamut may now render properly, highlights that may have been difficult to roll off smoothly might suddenly be easily finessed.

This also means that it is much easier to adapt old footage shot on raw to new delivery scenarios. You may have shot something in 4K with HD TVs in mind. If there is an opportunity that arises to deliver that in 4K HDR, you can go back and color grade it with HDR in mind, because the initial capture was more than sufficient for HDR.

## Raw Adjustments vs Normal Color Correction

It is vital to understand the difference between a raw adjustment and an ordinary color correction. Raw workflows give you the ability to adjust the ISO and white balance in post, just as if you were on set. This means that you can lower and raise the ISO without crushing the blacks or clipping the highlights. You can more easily compensate for spill from a green screen, or even crappy fluorescent lights that you'd find in an office. The difference between adjusting the tint on a 16-bit raw image vs an 8-bit h.264 image is jaw-dropping. All of that kind of information is just "metadata," and it can be easily tweaked to obtain your aesthetic goals.

## Financial Justification for Shooting Raw

Of course, raw data can take up more hard drive space than highly compressed data. You might be thinking about the increased cost of media, additional on-set storage, more active storage in post, more backup storage, longer file transfers, turn around speed, and so on. These are important considerations when evaluating the scope of your project. The good thing is that those are quantifiable costs. You will be able to calculate how much the extras space and time will cost you. Much of this pain has been eased by recent technology, however. You can now obtain a 10TB Thunderbolt 3 drive at a reasonable cost. SSD raids over USB-C and Thunderbolt 3 are becoming more common. When you combine the efficiency of the best raw codecs, new I/O connections, inexpensive storage options, and powerful new GPUs, raw often does make sense. The image quality and flexibility on set and in post make a compelling argument that the gains outweigh the costs in most filmmaking scenarios today.

## Conclusion

Raw presents the highest quality and most flexible option for filmmakers. And that's why many of the highest end Hollywood pictures all the way to independent documentaries, choose raw for the acquisition of their imagery.

Contributed by Reuben Evans, Executive Producer, Faithlife TV.

Because the decisions about the camera, codec, and color space affect so many stages of post-production, it's very important to have everyone on board with those decisions. The Director of

Photography, the DIT (digital imaging technician), the colorist, and the VFX houses should all be bought in.

### Why Do People Sometimes Use External Recorders to Capture Footage?

One way that people have found to capture higher-quality codecs on cheaper cameras is to use an external recorder.

Many prosumer cameras can offer excellent image quality but don't have the ability to record high-end codecs, and so it's common to pair a camera with an external recorder, giving you both the image quality and the recording format of a high-end camera, albeit in a more cumbersome package.

These devices (many of which can double as external monitors) take an uncompressed signal from the camera, via HDMI or SDI, and compress it separately. So you end up with two copies of your footage – one copy heavily compressed on the camera, and a second copy lightly compressed on the external recorder. The key thing here is that the camera sends the signal out to the recorder before compressing it.

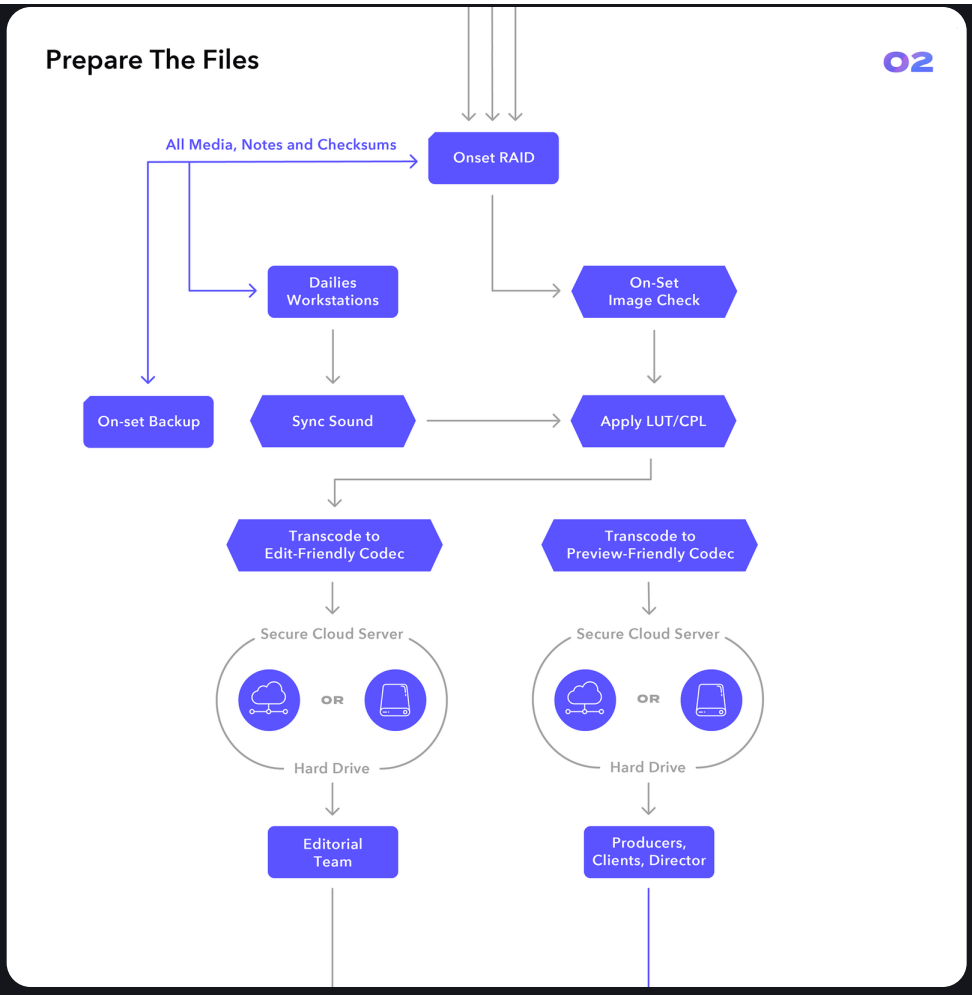
One important note here is that many cheaper cameras only output 8-bit, and often not in 4:4:4. An external recorder might be able to compress to a 12-bit codec, but if the camera is only sending 8 bits, the recorder can only record 8 bits. Some cheaper cameras may also not output a "clean" HDMI signal that is suitable for recording. We call an output signal "clean" when it's just the pure image with no camera interface overlays.

CHAPTER 02

# Prep

A lot has to happen between the time when the director yells “cut” and the editors begin their work. Syncing, backups, checksums, and more.

*8 minute read*




Whichever camera and settings you choose, the result of all your hard work from the shoot is going to be a memory card or SSD full of files. Probably many of them.

The very first thing you should do as soon as a card has been removed from the camera is back it up. In multiple places. At the end of the day, all of the results of your entire team's efforts are contained in a few very fragile pieces of metal and plastic.

Because those files are so precious, the DIT or camera operator will use special software to run the backups, rather than using the standard file copy functions. This software uses a checksum, which is like a digital fingerprint, to make sure that the files on the backup hard drives are *exactly* the same as the files on the memory cards.

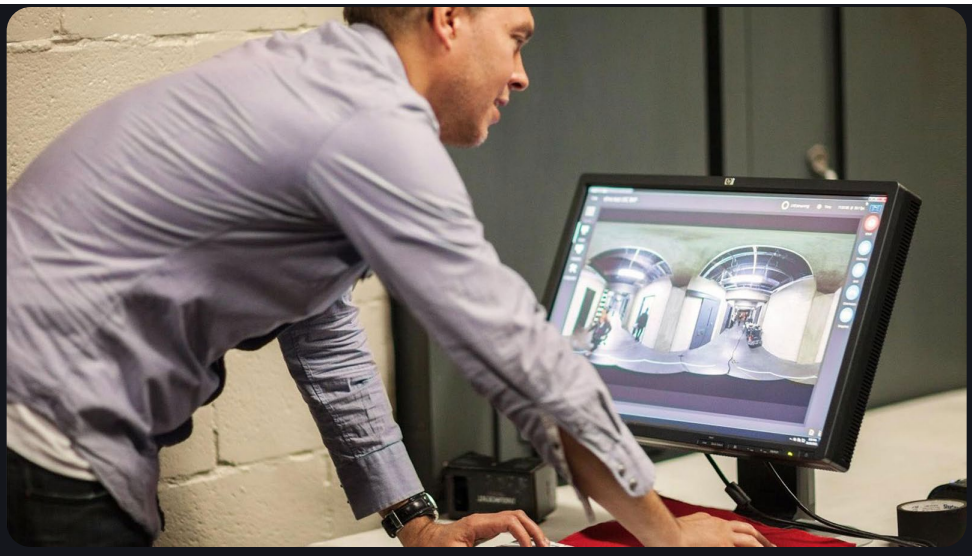


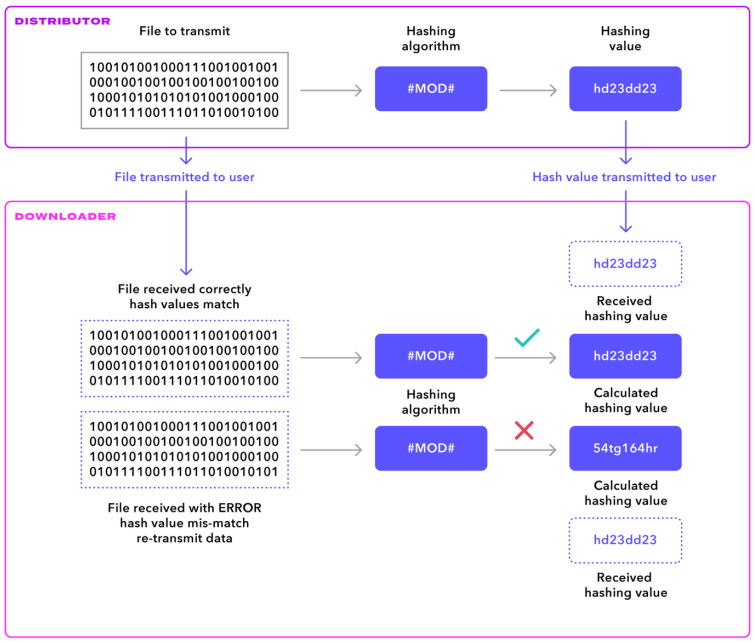
Image © Logra Studios Photography from the set of “The Suitcase”. Dir. Abi Damaris Corbin

Checksums and Robust Backups

## Checksums

If you are ever responsible for moving files around a workflow (perhaps offloading camera storage to a computer or transferring data between facilities) you should be using checksums to protect your assets.

A checksum is a unique piece of data that identifies a file. If even a single bit of data changes within a file, the new sum of the file will not match the original, and the checksum will know something went wrong. Because different files contain different data, every file has a unique sum, meaning you will always know if a file is 100% authentic/complete after you receive it. So, checksums are like fingerprints. They follow files wherever they go, and tell you if those files are what they say they are (or not).




Checksums can be used for any type of digital file (or even for collections of files), but they are particularly useful in post-production. They help safeguard your footage as it's handled by dozens of different users and let you know when there is a problem.

Because video files are made up of millions and billions (and sometimes trillions) of individual bits, handling them is a complex process. And complexity breeds errors. If just a single one of those digits gets corrupted or lost, then your files may be completely useless. For video production, losing files often equals losing hundreds or thousands of man hours of work. Lost footage from just a single take can cost a production tremendous amounts of money. While checksums won't protect against everything (like a drive being stolen or a nervous PA spilling coffee on the DIT cart), they are an important part of a safe workflow because they allow you to notice quickly when something has gone wrong.

Any time you are offloading footage from a day's shoot, creating backups, delivering dailies, or sending footage to

another team (basically any time you are moving files between machines), you should be using checksums to validate data integrity. While incorporating checksums into your workflow does require a small investment, the value they add in terms of safety and security is well worth it. Checksums detect problems as they happen, which lets you act immediately. This could potentially save you a lot of money (and countless grey hairs).

The actual mathematics behind checksums is pretty complicated (if you know what a hash is, a checksum is the same thing). Different algorithms use incredibly diverse methods to generate checksums for files, but in general, the process is the same. Let's start with a simple example.

Imagine a file as a number, for instance, 2341842. Because the data of this number is so simple, we can manually create a checksum by adding the digits of the number together.

$2+3+4+1+8+4+2=24$

Now that we have the checksum value, we can just tack it on to the end of the number, so the number is now 234184224. Since we did this process ourselves, we know the last two digits are not part of the original data, but are merely the checksum value.

Now imagine we want to send this file in the mail to a friend. Unfortunately, the letter got slightly wet along the way, and the 8 got smeared and now looks like a 6. This is a problem. However, the good news is that we also wrote in the letter that the last two digits of the number are the checksum value. So, when your friend reads the letter, they will know to add the digits together, and will immediately see that there is a problem.

$2+3+4+1+6+4+2\neq24$

This is a very simple checksum algorithm, but it gives you an idea of how data verification works. Obviously, real computer files are far more complex, so computers employ much more rigorous and foolproof methods. For example, the popular checksum algorithm Md5 uses a string of 32 letters and numbers to generate a checksum value. That makes Md5's validation process very precise. It is *extremely* unlikely that a file could change but still have the same checksum value.

To use checksums in your workflow, you will need software that supports them as a feature. By default, Windows and macOS do not check for bit-level errors, so you will need specialty software. Shotput Pro and Hedge are popular options, though more mainstream software like Davinci Resolve have now integrated checksum validation. If you know how to use the Terminal/Command Prompt, generating checksums yourself is actually quite simple. But no matter which software you use, the basic steps are the same.

First, the software reads the file and generates a checksum. This is known as the *source checksum*, and it is attached to the file. Next, your computer makes a copy of the files you want to move, and then transfers the copy to another location. Once the file copy has been fully written to the new location, the software reads the checksum of that file, which is called the *destination checksum*. Finally, the software compares the source and destination checksums to make sure they match exactly. If they do, the transfer was successful. If they don't, the software will sound the alarm.

If you're storing a hard drive on a shelf for a long time (which can result in small bits of informationg being lost), then it's a good idea to generate checksums for the files at the start, so you can double-check that your files are intact later on.

Other than picking software, you will also need to decide which algorithm you want to use. Different algorithms have

different inherent qualities/limitations, but most software allows you to pick according to your needs. As mentioned above, Md5 is very popular in the video industry, in large part to its speed and cross compatibility with Windows, macOS, and Linux. However, Md5 is pretty old, so many people are using newer algorithms like xxHash, SHA-2, or SHA-3.

If this seems complicated, don't worry, software makes the process mostly automated. In normal circumstances, using checksums is not that much more difficult than the traditional "drag and drop" method of file transfer. So really, there aren't many reasons not to be using checksums in your workflow. They offer peace-of-mind at very little cost to your time or budget.

However, as useful as checksums are for ensuring data integrity, they are only part of the solution to securing your video files. The bigger piece of the puzzle is building a robust backup system.

## Robust Backups

Not backing up your video files is like playing Russian roulette with your job and reputation. You might think your odds of survival without sufficient backups are good, but even just one lost file can lose clients/end contracts. With that in mind, you need to make a plan for backing up all of your data.

There is no one-size-fits-all scheme for backing up video files. Every workflow will have unique requirements that change the level of backup you need, but there are always limited resources to go around. The key task is to balance security and probability. You don't want to lose data, but you also don't want to spend your whole budget securing your footage. So, there are a few general rules you should follow when planning your backup system.

Rule 1: You should always (always) have at least one extra copy of every single file in your project. Everything from the most important shot of the entire film, to seemingly-inconsequential metadata files for the credit sequence. If someone in the workflow needs it, then it's important and should be backed up.

Rule 2: Backups must not be connected to the originals. Putting a copy and original on the same computer or network is a recipe for disaster. One virus or bug could delete both files. If the files are directly connected by a wired or wireless connection, then it's not a sufficient backup.

Rule 3: Backups should be in a different physical location than originals. Power goes out, thieves break in, and moth and rust destroy. When disaster strikes your office or studio, all your computer systems and storage could be ruined. The best way to prepare for this issue is to have offsite backups.

Rule 4: All backup storage must be independently secure and stable. This rule dictates what type of storage you should actually use to back up your critical files. Sure, you might have terabytes of files in offsite backups, but if all that data is being kept on 10 year-old hard drives, then those files are probably not that safe. The same can be said of certain RAID configurations that have a relatively low fault threshold. While you might think RAID arrays are generally more secure than single drives, this is not always true. RAID 0, for example, provides no redundancy for your files, so if even one disk fails, you lose everything. RAID 0 is actually *less* stable than a single hard drive. Always keep your backup files on the most reliable, durable storage media you can find.

Rule 5: Backups must be organized. If you follow the above rules, but don't keep a detailed record of where you put your backup files and what they are called, then you might as well

not have bothered. If disaster strikes your production, and you need to pull your backup files, it is absolutely necessary that you know where everything is. Else, the backups are practically nonexistent.

Rule 6: Backups must be accessible. Just like the previous rule, backup files are no good unless you have them in hand. You might have an offsite backup in a safety deposit box somewhere, but if the bank isn't open 24/7, then you may not be able to restore files in time to fix a problem or meet a deadline. Even if you keep offsite backups in the cloud, a slow internet connection can get you in trouble. You might have to wait multiple days to download your terabyte-sized backup. This is not to say *all* of your backups must be accessible at all times, but at least one complete backup should be.

Following these rules will make your backup procedures much more robust. Of course, there are always risks, but you can do your part to mitigate them.

Later, you will very likely convert these files into different formats, but you should *always* keep the original camera files carefully backed up. These files are also sometimes called the "camera negative" (a reference to the days of physical film cameras).

## Dailies

Dailies (also referred to as "rushes") is the collective term for all of the footage that was shot on a given day. At the end of the day (and sometimes also in the middle of the day), the DIT preps the dailies and sends them off to the producers to view, and to the editors to begin creating assembly edits.

The DIT can't simply copy the files from the memory card to a hard

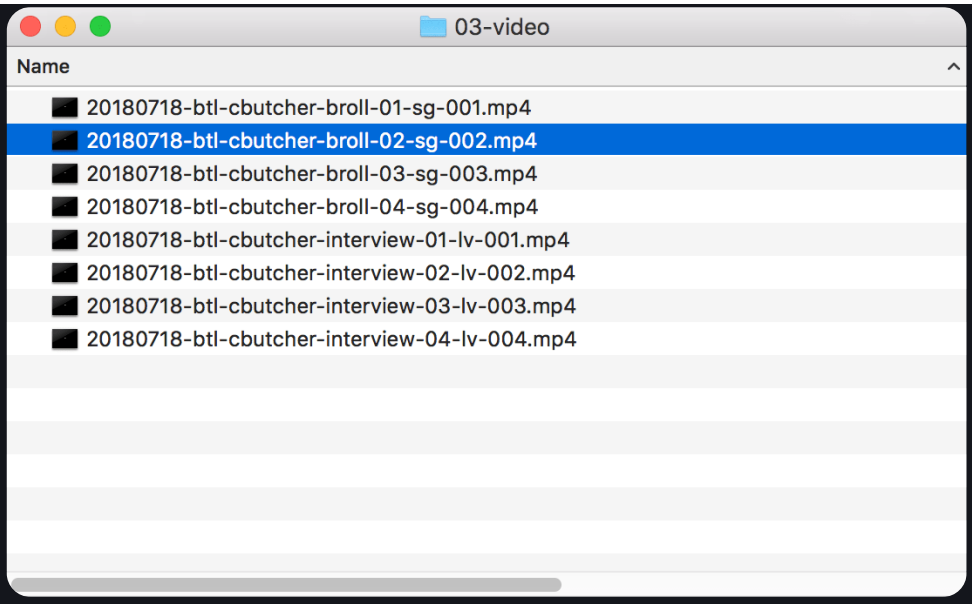
drive and hand it off, however. Several important tasks must happen first.

## The footage must be organized

Producing a feature film will involve tens of thousands of individual audio and video files, created over many weeks, and it's extremely important that all of those files are organized properly. The cameras do not do this automatically!

A carefully-maintained folder structure system helps keep each file in exactly the right place, and the files themselves will often be renamed so that, even if they get moved out of the correct folder, it will be clear where they belong.

| | |
|---|---|
| FILE NAME | **20180901-btl-cbutcher-interview-01-lv-001** |
| SHOOT DATE | **20180901**-btl-cbutcher-interview-01-lv-001 |
| PROJECT ID | 20180901-**btl**-cbutcher-interview-01-lv-001 |
| SUBJECT NAME | 20180901-btl-**cbutcher**-interview-01-lv-001 |
| SHOOT | 20180901-btl-cbutcher-**interview**-01-lv-001 |
| CAMERA NUMBER | 20180901-btl-cbutcher-interview-**01**-lv-001 |
| AUDIO | 20180901-btl-cbutcher-interview-01-**lv**-001 |
| CLIP NUMBER | 20180901-btl-cbutcher-interview-01-lv-**001** |




## File naming and organization

Before a single piece of footage is even shot, it is critical that you decide on a file naming convention for the entire project.

A file naming convention is a set of rules for assigning unique names to all your digital media files. Cameras, computers, and other recording devices have their own default naming rules, but they are generally just a sequential label.

WB2A2468.MOV

This is not a useful filename, because there is no way to know what it is unless you play it back.

A good file naming convention actually describes what a digital media file contains and how it relates to other files. These rules usually require important information like date of creation, project name, and various technical details.

20181004-wfg-filenaming-camA-01-001-4k30.mov

This is a much better filename. It gives the shoot date as YYYYMMDD, a project ID, scene name, source camera, shot and take numbers, and the format it was recorded in. Anyone can quickly understand what it is, whether they see it the day it was shot or many months later.

The exact structure of the file naming convention you need varies depending on the type of project and complexity of the workflow. For example, a narrative film will likely include scene and take numbers, while a broadcast program will list season, episode, and subject markers.

Narrative Film: 20180606-wfg-camB-roll03-shot05-take002-4k30.mov

Broadcast Program: 20180215-wfg-s02-ep04-bts-1080p60.mov

Whatever file naming convention you use, it needs to be clear and consistent from the very beginning. This will make your media management more efficient and reduce risk. For example, since good file naming rules forbid duplicate filenames, it is much more difficult to accidentally overwrite footage during camera offloads. Similarly, unique filenames decrease confusion during dailies prep and sharing.

But a well-planned file naming convention serves all files throughout the entire scope of a project, not just files coming off of a camera.

As media files move down the post-production pipeline, different teams will be handing off thousands of files constantly. Even simple deviations can lead to disaster. If the DIT uses one date format, while editorial uses another, and VFX doesn't include it at all, then media files can easily be lost.

This is why everyone involved in the production needs to test

and agree on the file naming convention before the project gets underway. A suitable set of file naming rules should be robust enough to meet the needs of every team, but simple enough that they can be used easily.

You will also need a plan for organizing files into folders/directories. There are a huge number of ways to set up a folder hierarchy, but in general, it should logically guide users to existing files, and indicate where new files should be put. It is important to make sure your organizational scheme can grow with the project, from the first 100 pieces of footage to the last 100000 VFX, color, and sound files, and everything in between.

Ultimately, file naming rules and directory structures are all about keeping your media organized and secure. If you are proactive about this seemingly mundane technical detail, then your team can focus more on the creative aspects of your project. You will have to invest some time and effort to do it right, but it will pay big returns down the line.

## Sound must be synced

In order for the producers and editors to properly review dailies, the DIT or dailies team must carefully sync the audio and video together.

Even though all digital cameras are capable of recording sound and video together in the same file, it's still very common for the film's sound to be recorded separately. This is called dual-system sound.

If you're going for the highest possible sound quality, then you'll get

a better recording if you use a dedicated sound recorder. The sound recording capabilities of most cameras, even professional cinema cameras, can't match a high-end sound recorder.

And then, just logistically, it often makes sense for the sound to be recorded separately. The sound recordist is going to be carefully monitoring and tweaking the sound levels during recording, and you don't normally want an extra person huddled over the camera pressing buttons while the camera is rolling.

Syncing sound

There are three ways to ensure that the audio and video are matched properly: slating, timecode, and scratch audio.

It's very common to use two of those methods (or even all three) simultaneously to ensure that, if one of them fails, the sound doesn't have to be synced by eye, which is a *huge* headache.

## Slating

Slating is the most well-known system, because it goes back to the very beginnings of filmmaking. The iconic film slate is a chalkboard that has the current scene, shot, and take number written on it.

When the actors are ready, someone (often the 1st Assistant Director) will say "roll cameras" and "roll sound." Once everyone is ready, an assistant will read out the information written on the slate, and then clap down the clapper to make a loud noise.

Later on, when you're combining the separate audio and video files, you can easily pair the right two files together by matching the scene and take information in the video and audio. The clap of the clapper provides a clear and precise point at which to sync the two files.

As I'm sure you can imagine, syncing sound via a slate is still a somewhat tedious process, but it's fairly straightforward and reliable.

## Timecode

Basically, timecode is a clock that ticks in frames, rather than just seconds. It keeps track of when video and audio is recorded, and then stores that information as metadata within the file. When it comes time to edit the files, your NLE can match up all your audio and video based on this information.

At the beginning of a shoot, timecode is synchronized between devices. By doing this, all the footage and sound run off the same clock, which means that any particular event will have the same timecode across every device.

This system is not perfect, however. Different devices will have

slightly different clocks, so they will drift apart over time. That means they need to be resynchronized to each other periodically to keep everything aligned. You can also feed an external timecode source to every device on set, so that no drift occurs.

As long as the timecode sources are synced across all your recording devices, then matching audio and video should be a relatively automated process. It does take a little more work on set than just using clappers, but it saves a lot of time in post.

Deep Dive: Timecode

Contributed by Hilda Saffari, Editor and Post-Production Consultant.

## Scratch audio

We mentioned above that almost all digital cameras can record audio, although the quality might not be the best possible. A simple and effective way to sync sound and video is to record the sound both on the camera and on a separate sound recorder.

The waveform of the audio file may be slightly different between the two recordings, but they will be similar enough that software can usually match them up automatically.

If the camera's internal mics aren't close enough to the actors to hear them properly, the sound recordist may send a wireless signal out of his equipment into the camera in order to ensure that the two audio signals are similar enough.

Using scratch audio is one of the simplest and easiest ways to sync audio, because it takes almost no extra work on set, and it doesn't rely on any extra hardware.

It can still be a good idea to have another method as a backup, however. You may forget to turn the camera's sound on, for instance. Or your sound recordist may capture a ton of ambient noise that makes the recording unusable.

There are some situations (a sit-down interview, for instance), where separate sound recording is probably not necessary. In that scenario, it's unlikely that the volume of the sound will have huge changes, so the sound recordist doesn't need to be tweaking dials while the camera is rolling. And if you're using professional mics and a professional camera, you may be perfectly happy with the audio quality. It's all relative.

Interview with Emmy-award-winning Imaginary Forces Creative Director Karin Fong.

## Metadata must be handled

Metadata is data *about* a video file that goes along with the file, separately from the video and audio itself. This data can be extremely valuable for the rest of the post-production process. Some metadata (e.g. shutter speed, lens information, date, time, timecode, etc.) is captured by the camera itself. That lens information is very valuable if you are doing any CGI, for instance.

What Is Metadata?

## What is Metadata

You've almost certainly heard the term metadata and probably have some idea that it means information about frame rate and codecs. That's true, but metadata is much more. If you break the word apart into "meta" and "data". Meta literally means "about the thing" and "data" is the thing.

In the world of production, the "thing" is referred to as the "essence," i.e. the media itself. Everything else is metadata. Everything we know about the media is metadata, whether it's technical information or information about the content, everything we know is metadata.

Metadata is not new; it's just a new name for something you're probably already using. Instead of calling it metadata you're much more likely to be calling it "notes", "log notes", "script notes" or "logging".

Notes about the script, location notes, script notes, tape or card labels, log notes, and even transcriptions are additional descriptions relating to the shoot and edit that aren't the media itself. They are all metadata.

This isn't anything new, but the digital world lets us do so much more with metadata than we could with paper notes. That they are searchable alone make a huge difference.

In our media-centric world, we have technical metadata that describes the technical aspects of the file, such as frame rate, image size, codec, etc. Technical metadata is easy to derive because it's inherent in the characteristics of the file.

Technical metadata applies to the entire media file (and all clips derived from it) because from the technical perspective, every frame is the same. Technical metadata tells us nothing about the content of the file.

Content Metadata describes the contents of the film or video image by applying a description of the content to all, or part, of the file. Without the Content Metadata, we know nothing about the content of the file. Content Metadata can apply to only part of a media file because the description will change as the content changes. An example of a piece of Content Metadata might be "Close up", but if the camera moves, it might become a "Medium shot" within the same clip.

Metadata doesn't just apply to our media world. Metadata is being gathered as we watch videos on YouTube and rate them, as we make purchases online, and as we make phone calls. You will find more on the connection between metadata and the uses discussed on the news in Metadata and the Wider World.

In the words of filmmaker and archivist Jason Scott's, "Metadata is a love note to the future."

As we alluded to above, in post-production there are two primary types of metadata: technical and content. There is a third use of metadata in the media and entertainment

industries, and that's distribution metadata covering the content and rights associated with a finished program or movie.

## Technical Metadata

Technical Metadata covers the range of information we get from the camera (generally). Not surprisingly, it is the technical information about the media file. This includes the obvious technical information on Image size, Frame rate, Codec, etc. But technical metadata goes beyond the obvious and – depending on the camera or source – can include focus, white balance, GPS, camera and lens serial numbers, and a lot more metadata than you would think necessary.

Technical Metadata is used to manage media and to automate operations.

Technical metadata comes with the file from the camera because it is dependent on the camera and the type of file(s) it produces. Some technical metadata is in the file itself, but in other cases, it resides in a separate "sidecar" file in the same directory as the media file. Sidecar files are usually a XML files with relevant information in them. Each type of XML file is specific to the camera that created it.

These sidecar files are why it is important to always maintain the entire card structure before importing to an NLE.

## Content Metadata

Content Metadata is the part of the metadata spectrum that is most useful in production. This is information about what is in

the media file. Once it was obvious (though slow) to find out what was in a frame of film by holding the film up to the light, but now metadata about the content of the media file is how we understand what we have.

Content Metadata can be as concise as a Keyword or as verbose as a transcript. Content metadata is generally – for the moment at least – added by a human somewhere before editorial starts. We say "mostly" because increasingly indexing speech and images for searching can be done by trained machines, sometimes inaccurately referred to as "Artificial Intelligence."

Machine-based logging and metadata generation has its limits because, at the current state of the art, there is no discrimination about what is useful metadata and what is simply cluttering up the index!

Maximum discoverability makes sense in an asset library, where vast amounts of video are stored, but the flood of metadata out of machine-driven indexes often becomes too overwhelming to be useful in production. Content Metadata for production needs to be focused on the needs of the production and, aside from basic image search and transcripts, likely to be too broad to be useful.

Content Metadata is what we use for organizing projects and getting a head start on editing.Content Metadata is usually applied and managed within an NLE. Content Metadata generally does NOT travel with the file, but rather in the NLE's Project file(s).

Content Metadata does not need to apply to an entire media file. It is more valuable when it is applied to content-based 'selects' or defined time ranges.

## Distribution Metadata

Distribution Metadata solves a problem most of us don't have: tracking the distribution of finished programs through their many territories and variations. The industry solution is EIDR: Entertainment IDentifier Registry. It is a unique ID for every variation of every film and television episode ever made.

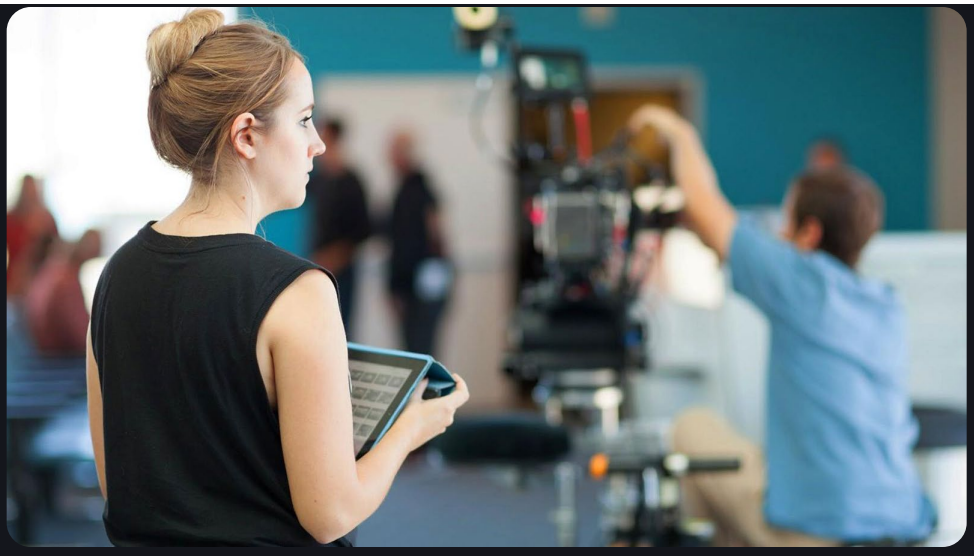
Because it is unique, the EIDR can be used by third parties – most notably ROVI – to add additional value for program guides by linking the program/episode ID to other data sources.

Contributed by Philip Hodgetts, Intelligent Assistance.

Some metadata is recorded separately (often by a script supervisor), and might include the shot, scene, take information, the actors who appear in the shot, and the director's comments about that shot. If the editors (who usually were not on set) know which takes the director really liked, that will help them build an edit that the director will be happy with.

Script Supervisor on the set of "The Suitcase" with iPad app for logging scene and metadata. Image © Logra Studio Photography. Dir. Abi Damaris Corbin

Keeping track of all of that data is a surprisingly complex task, especially since different software handles metadata differently.



Managing metadata

How you handle metadata through your workflow depends on the type of metadata we're dealing with. Technical metadata always relates to media files and travels with the media file itself, because technical metadata is always about the file itself.

Content Metadata does not live with the media file. It generally lives in a Project or Library specific to the NLE that was used to enter log notes, keyword ranges, and other metadata about the content of the clips.

## Handling Technical Metadata

Because Technical metadata travels with the file, there's very little work involved in maintaining the technical metadata until you re-encode a file. Until that point, file name and timecode (in particular) were constant. When we re-encode a file, the file name may be lost, modified or in a specific file path (to avoid naming conflicts). Every re-encode changes the creation date and generally also changes the file size and codec.

This becomes important when we're working with a proxy workflow. Tracking the relationship between original/source media and the proxies used for editorial on feature films has been big businesses for companies like Deluxe and Technicolor. Traditionally, the preparation of proxies, tracking relationships, and matching back to original media after editorial is complete has been outsourced to these companies, or companies like them.

This is still likely to be the workflow for most feature films, particularly those edited with Avid's Media Composer. It's also the least stressful workflow for editors and allows a wide selection of codecs or frame sizes for the proxy media.

Final Cut Pro X and Premiere Pro CC both have native proxy workflows, where proxies are created and managed by the host NLE. This also has low stress for editors, but does not provide much flexibility on what is used as a proxy.

For those who want control and want to manage the process themselves, or internally to their organization, Blackmagic Resolve provides a lot of flexibility in forcing relinking.

## Handling Content Metadata

Content metadata does not live with the file. It exists only in an NLE or Media Asset Management System (MAM). This makes the connection between clips and content metadata rather fragile. In most cases, simply copying the media is not going to bring the logging associated with that asset or clip.

What makes this more complex is that there is no universal method of handling Content Metadata. The way the information is stored in each NLE is proprietary to that NLE. Projects from Premiere Pro are structurally different from

Media Composer projects, Resolve projects or Final Cut Pro X Libraries. Because of these different structures, it's not possible to do a complete translation from one NLE's Content Metadata to another's structure.

For this reason, it is always best to remain within one NLE from beginning to end of the project. The exception is going from any NLE for editorial, to Resolve for color grading. While that is generally a smooth transition, there's no real focus on retaining Content Metadata because Content Metadata is particularly useful in the editing process.

Adobe does embed metadata added to the XMP Metadata Panel into the media files. Once embedded, another install of Premiere Pro CC will read that metadata from the file and make it available in the XMP Panel.

Something similar can be achieved for FCP X using Intelligent Assistance's FindrCat. FindrCat embeds all the logging information from FCP X into the media file and translates Keywords to Finder Tags. The full metadata can be returned to FCP X from the file using the Pro version of FindrCat.

Another way of storing and managing Content Metadata is using an asset management system known as Digital Asset Management (DAM) or Media Asset Management (MAM) systems. These tools read in Content Metadata from NLE's via XML and store it within the DAM or MAM in their own data structure.

Most asset management systems support multiple NLEs, so asset metadata can be read into the MAM from one format of XML and clips can be exported with metadata to another format of XML, making an MAM or DAM a means of managing metadata during post-product

Contributed by Philip Hodgetts, Intelligent Assistance.

## The footage may need to be transcoded

The files that come straight out of the camera are usually not well-suited for editing. Depending on the camera and codec, they may even be impossible to play back on a normal computer. So, before anybody views the dailies, they often must be transcoded.

Transcoding is the process of converting files from one codec into another, and the DIT may even have to transcode the files into multiple formats–one for the producers/director to view, and another for the editors to use.

The editors may request a specific codec. Avid's DNxHR and Apple's ProRes are very common for editorial. In our workflow analysis of the 2018 Oscar nominees, we found that the DNx 115 codec was extremely popular, in spite of its age. If the editors are using Final Cut Pro X, however, they may request a ProRes codec. While DNxHR codecs are more common among feature films, ProRes is more common in broadcast TV.

[Check out the Codec Comparison tool to see how different codecs impact image quality and playback performance.]

### Editing Codecs

When choosing a codec for the editing phase of your project, there are some important considerations to keep in mind.

## Compressed Codecs will Slow Down Your Editing.

Most lower to mid-range cameras record with codecs that use temporal compression, also known as long-GOP (group of pictures) compression. These codecs do a great job of minimizing the file size of your footage, but your computer will have to work much harder to edit them. Why is this?

The simple explanation is that temporal compression saves incomplete information for every frame. Long-GOP codecs track the video from frame to frame, and then only save the information that changes between them. Anything that stays the same is merely referenced from the frame at the beginning of the group. This means that every frame is not a complete picture, but rather just the "new" information necessary to reproduce changes in the image.

If a video doesn't include a lot of motion, then the video file can be a whole lot smaller than it would be otherwise. For example, imagine a video where the difference between the first frame and last frame is just a few pixels. In this case, the codec only needs to store the identical pixels once, and the "new" pixels every time they change between frames.

If this sounds great, try editing with a long-GOP codec. As you scrub through the timeline, your computer will likely stutter at least a few times. That's because the computer has to look at a whole series of frames to reproduce an image. If you're

playing the footage forward, that's not usually an issue, but when you're jumping around a sequence or even playing a clip backward, there will be a high computational cost. This can slow your editing down significantly, especially if you're editing in 4K.

When transcoding your files for editorial, be sure to avoid long-GOP codecs (like most versions of H.264). The better option is to choose an Intraframe codec, which stores a complete image for every frame. File sizes for intraframe compression are obviously bigger, but they do not incur the same computational burden of Long-GOP codecs. ProRes, DNxHD/HR, and Cineform are three great codec options for editing.

## High Bitrate Codecs Will Slow Down Your Editing

For low to mid-range codecs, you don't have to worry about the bitrate at all. Once you start moving up the codec ladder, however, high bitrate codecs can cause issues with editing, especially if you're working on "everyday" computers and high resolutions.

The reason is that your computer needs to be able to read the data from your hard drive at a rate that is at least as high as your codec's bitrate. It makes sense – if your codec has a bitrate of 50Mb/s (fifty megabits per second), then your computer needs to be able to read that file from the hard drive at least that fast. Else, the playback will fall behind and stutter.

[Note: Mb/s stands for megabits per second, while MB/s stands for megabytes per second. There are eight bits in a byte, so you need to multiply by 8 when converting from MB/s to Mb/s]

The good news is that hard drives are getting faster every day, so a bitrate of 50Mb/s is not going to cause any problems. But what if you're editing ProRes 422HQ at 4K, which has a bitrate of 734Mb/s? The average external hard drive is only just barely fast enough to play that back, and some cheaper hard drives won't manage it at all.

And then, what if you're editing a multicam sequence with three cameras? Now you need 3x that data rate: 2,202Mb/s! At that point, you're going to need to invest in some higher-performance storage. If you don't, then your timeline is going to be pit of despair.

Some editors already have such speedy storage. However, if you do not, you can achieve significant savings both in cost and flexibility by using a proxy with a manageable bitrate. If you go that route, it may be feasible to keep the entire film stored on the internal hard drive of a laptop, allowing you to work from the beach, from home, or on a plane or train.

## Editing in LOG will Slow Down Your Editing

Shooting in LOG is one of the best ways to preserve as much of your image's dynamic range as possible. This gives you much more flexibility when it comes to crafting the "look" you want in post. Now that LOG profiles are available even on most prosumer cameras, it's an extremely popular way to shoot.

The downside is that LOG images that come straight out of the camera don't look so great. It can be difficult to pick the best take from a scene when you can hardly distinguish an actor's facial expression from the wall behind them. So, you need to add in a bunch of contrast and saturation in order to preview the image accurately and make your creative

decisions. The easiest way to do that is to apply the appropriate LUT to all of the clips when editing.

This can be a huge annoyance to manage, and can also slow down the computer. With a LUT applied to every clip, that adds another computational step. First the computer has to decode each frame, then it has to apply the LUT, and only after that can it display the image. Of course, you can avoid this issue editing the LOG footage without a LUT, but that's certainly not ideal.

The most common way to overcome this issue is to pre-bake the LUT into proxies files of the LOG footage. If you're going to convert your files into proxies before editing them anyway, then you might as well apply the LUT during the transcode process. That way, the editor is always working with footage that has good contrast and color, but doesn't have to bother with LUT management or decreased performance. Keep in mind, you should only bake-in LUTs if you're going to switch back to the original camera files later.

Unfortunately, the files that editors like to use aren't as convenient for producers to review, partly because of their file size and partly because they may require special software to be installed. The DIT will probably also generate a *separate* set of dailies for producers to review in the H.264 codec, which are easy to play back on any computer.

If you're using a tool like Frame.io to collaborate online, then it's not necessary to generate separate dailies for the producers/editors, since Frame.io generates them automatically. Cloud-based tools are becoming more and more popular for dailies review because they make it easy for the team to collaborate from different locations, with the footage available from a web browser or mobile app.

Executive Producer at Ingenuity Studios reviewing VFX shots in Frame.io from the movie "Get Out."

## LUTs must be applied

In the section on log recording, we learned that most video is captured in a very low-contrast, low-saturation format that isn't very nice to look at. In order to preview these files properly, we must add saturation and contrast (also known as gamma) back in.

The most common way to do that is by applying a LUT to the footage, which automatically converts the footage into a more natural-looking image. These "technical LUTs" are designed specifically to match the camera's sensor and settings.

### LUTs and CDLs

Let's talk a bit about LUTs and CDLs here and explain what they are and where they are being used in production. The main question is when to use one or the other and what the

advantages of each one are.

## LUTs and how they work

Starting with the most known and used – the LUT, or Lookup Table.

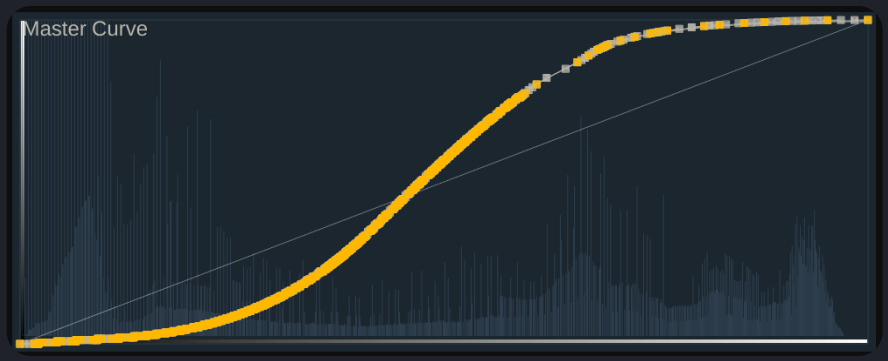
iRGB.lut

```
#
#       Example Look-Up Table (LUT) file:
#       ---------------------------------
#       Mapping an image through this colourmap LUT creates its
#       negative.
#
#       Note use of the "LUT:" keyword below.  Format is:
#               LUT: <number of tables> <table length>
#       A number of intensity values equal to "number of tables" X
#       "table length" is expected, one-per-line, following the
#       "LUT:" keyword.  Each value must be in the range 0-255.
#
#       If there is only one table, each of R,G,B are mapped through it.
#       If there are three tables, R is mapped through the first,
#       G through the second, and B through the third.
#       If there are four tables, R is mapped through the first,
#       G through the second, B through the third, and A through
#       the fourth.
#

LUT: 1 256

        255
        254
        253
        252
        251
        250
        249
        248
        247
        246
        245
        244
        243
        242
        241
        240
        239
        238
        237
        236
        235
        234
        233
        232
```

A Lookup Table basically works like a digital camera filter– meaning that it is a primary adjustment. It affects the entire image–a LUT cannot reflect keyers, power windows or any kind of tool that isolates the adjustment to a part of the image.

The effect is similar to just adding a blue piece of glass in front of the camera when shooting: The glass turns the image blue and the effect is definite – meaning that if you want less of the

blue effect, there is no way of making the glass "less blue". It's similar in your software of choice–it offers you for instance a slider that lets you dial in a blue tint dynamically – however, a LUT you can either load, or no –you cannot load a LUT only half way. If you want to mitigate the effect of the LUT, you need to dial against it with another tool in your software– for instance, load the LUT onto a layer (or node) and dial down that layer's/node's opacity.




That however makes it a bit of a double back and forth: Add 50, subtract 10 not an ideal process. Therein lies a big advantage of a CDL, but we'll get to that in just a bit.

There are usually two kinds of applications for LUTs: One is technical LUTs, where the LUT is supposed to do a technical conversion from one color space to another, without adding any kind of creative look into the image. Such a conversion can be for instance RED Wide Gamut to Rec709.

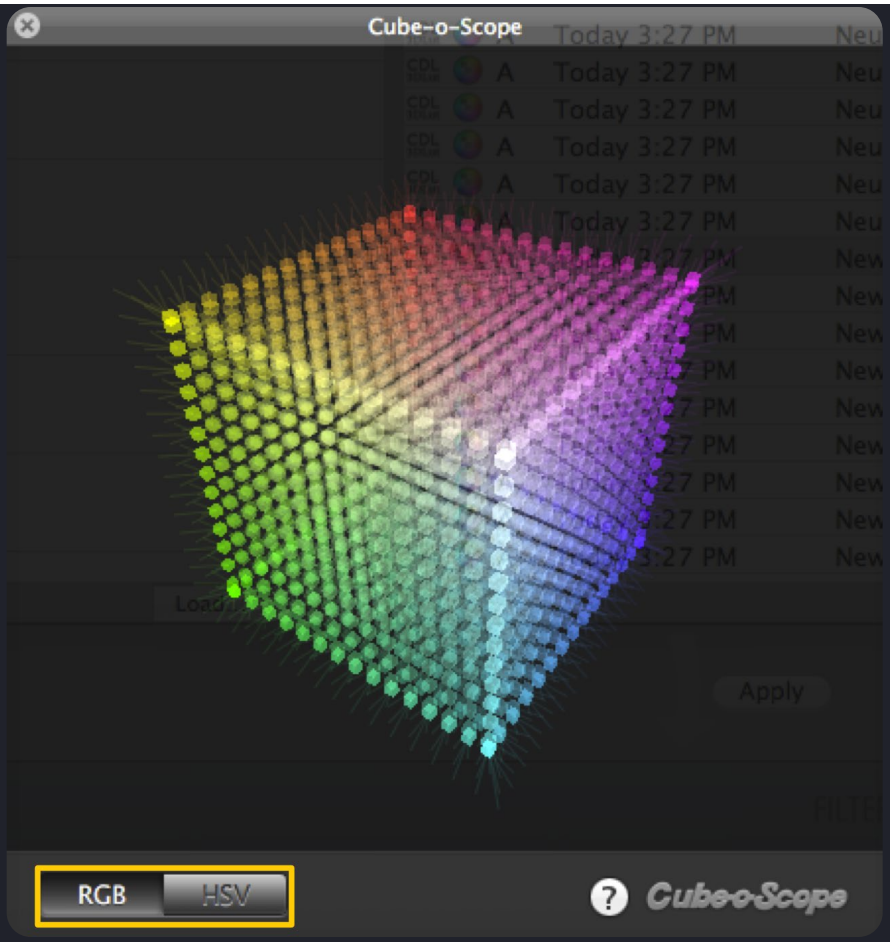
Another example is the use of calibration LUTs, where the only purpose is to calibrate the corresponding display in order to show reference colors and light levels.

The other application for LUTs is obviously the creative field. There are lots of LUT-packages out there (as anyone can create them easily) which define certain looks – cold, warm, vintage and whatever you want to call them. Creative LUTs can serve as a starting point for a creative grade. Swapping out a LUT against another one is as easy as changing a glass filter in

front of a camera lens, hence it is perfectly suited for quickly shuttling through a number of different looks for a given scene.

Another advantage of LUTs is that they can be loaded into a multitude of hardware devices, like Monitors and LUT-boxes, that can process it on the fly without further delay.

Creative LUTs are a good way to transfer the intended look of a given scene from set to post: During the shoot, the live SDI signal from the camera is being routed through a LUT-box, which modifies the live SDI signal on the fly – being controlled by any software that supports the LUT-box–most established is Pomfort's LiveGrade in this field. The look dialed in for a particular scene can then be saved as a LUT and forwarded to post as a starting point for the final grade. This LUT can also be used to create the offline dailies, that are going to be used in editorial: This way there is already a rough look applied during the edit.




An important thing to understand about using creative LUTs in post is that they don't free you from actually grading and balancing your clips. Let's say you have two clips of the same scene – one shot in bright sunny daylight, and when the second one got shot, there was a cloud in front of the sun and the whole image got much darker.

If we now apply the same LUT to both clips, the effect will be the same (let's go with the blue-ish tint here again)– but still one clip will show darker than the other, so balancing the two, so they match each other, is inevitable.

Another thing to keep in mind is that low-resolution LUTs can indeed cause banding and artifacts in your image, so don't trust just any free LUT-package out there that comes across.

LUTs exist in two flavors: 1D and 3D

KD5207Vis3_custom02.cube

```
# Generated by FilmConvert
# Date: 2016/01/21
# Time 17:32:19
TITLE "KD5207Vis3_custom02.cube"
LUT_3D_SIZE 64

0.000000 0.045082 0.095975
0.000000 0.045082 0.095975
0.000000 0.045082 0.095975
0.000000 0.045082 0.095975
0.000000 0.045082 0.095975
0.000000 0.045082 0.095975
0.000000 0.045082 0.095975
0.000000 0.045082 0.095975
0.000000 0.044524 0.095407
0.000000 0.037568 0.088093
0.051136 0.030828 0.080974
0.110269 0.024294 0.073912
0.168585 0.017971 0.066895
0.224040 0.011799 0.059950
0.275634 0.005864 0.053080
0.323379 0.000183 0.046208
0.367003 0.000000 0.039469
0.406809 0.000000 0.032859
0.442987 0.000000 0.026357
0.475649 0.000000 0.019986
0.505020 0.000000 0.013823
0.531720 0.000000 0.007846
0.556585 0.000000 0.002116
0.580201 0.000000 0.000000
0.602926 0.000000 0.000000
0.624904 0.000000 0.000000
0.646353 0.000000 0.000000
0.667283 0.000000 0.000000
0.687901 0.000000 0.000000
0.708172 0.000000 0.000000
0.728052 0.000000 0.000000
0.747804 0.000000 0.000000
0.767310 0.000000 0.000000
0.786642 0.000000 0.000000
0.805777 0.000000 0.000000
0.824807 0.000000 0.000000
0.843841 0.000000 0.000000
0.862730 0.000000 0.000000
```

1D LUTs are obviously the simpler ones, since they are basically just a gamma curve (and can easily be converted to one): An input value maps to an output value and that's about it. 1D LUTs are mainly used for log conversions (e.g. ARRI LogC to 2.4 gamma).

The most common file extension is .lut – sometimes (but rarely) you can also find them in an XML format. The accuracy of a LUT is expressed in two parameters: Size and Depth. Size defines the number of input values that can be mapped – the

more values between 0 and 1 (black and white), the more accurate the LUT is.

And Depth is the same accuracy, but on the output side. If you open up a 1D LUT with a text editor, size is basically the number of entries in there, and depth the accuracy of those entries: With a depth of 8 bit you will find values between 0 and 254, whereas with a depth of 10 bit you will find values between 0 and 1023 – a much finer grid.

A 1D LUT can contain just one table that maps R,G and B channels equally, or it can contain 3 tables to map each one and even a fourth channel for alpha (if supported by the software).

Now for the 3D LUTs – no, it has nothing to do with stereoscopic 3D. Instead, imagine a 3D cube, where hue, saturation and lightness are mapped to each axis: X, Y and Z. The same input/output game that we know from the 1D LUTs plays here in 3D space. Effectively, this means that you can e.g. map a red-color pixel directly towards green, whereas with a 1D LUT you can only increase, or decrease a red value. Same as with a 1D LUT, the accuracy of a 3D LUT is expressed through size (of the cube) and bit depth (of the entries inside).

The most common file extensions you will find are .3dl and .cube – but there are more, less common ones out there, too.

An important thing to keep in mind: Not all softwares and

hardware boxes support 3D LUTs equally – make sure you have your LUTs in the correct file format, size and depth for the software/hardware you intend to use them with.

## CDLs

So there's this other thing, called "CDL" – Color Decision List.

A004_C172_0901LE[1].cdl

```
<?xml version="1.0" encoding="UTF-8"?>
<ColorDecisionList xmlns="urn:ASC:CDL:v1.01">
        <ColorDecision>
                <ColorCorrection>
                        <SOPNode>
                                <Slope>1.113899 0.981387 0.964440</Slope>
                                <Offset>-0.097117 -0.045182 -0.003995</Offset>
                                <Power>1.140029 1.227524 1.350713</Power>
                        </SOPNode>
                        <SatNode>
                                <Saturation>1.500000</Saturation>
                        </SatNode>
                </ColorCorrection>
        </ColorDecision>
</ColorDecisionList>
```

A CDL is a standardized set of four parameters: Offset, Power, Slope and Saturation. That is obviously a very simple and limited toolset, but the upsides are compelling: A CDL is less destructive than a LUT, supported by all major grading- and editing- softwares out there and unlike a LUT, it is adjustable even after it has been applied.

What does that mean practically?

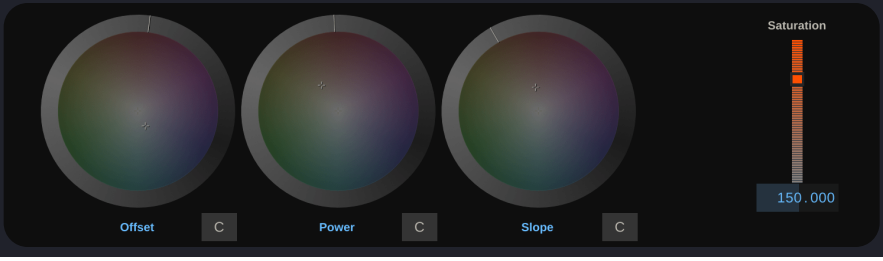
Let's stick to our on-set example here. During the shoot, the looks for each scene are created using a CDL toolset in our live grading software of choice. All looks are then exported in a CDL format: This can be .cdl, or (mostly used by vfx softwares) .cc. Many softwares also allow to export an ALE (Avid Log Exchange) that contains the CDL values per clip

and can be conformed in another software, so all looks can be applied in one go. Also a good old EDL (Edit Decision List) can, in addition to the actual edit, also contain the CDL values per clip. To be complete, there is another alternative for a format that can contain multiple CDLs: The .ccc format, which only contains a number of separate CDLs – this is however rarely used as it's lacking a lot of metadata, which is often required. Most of the time when receiving a .ccc file, there is actually only one single CDL grade in there anyways.

Now here comes the interesting part: When a CDL gets loaded into a grading software, the CDL uses the built-in CDL tools of our software to achieve the look. This means, if we store a CDL, which increases the saturation from 100 to 150 and load that CDL into our grading software, we will actually see our saturation slider go up to 150. If we decide to dial back the saturation, we can just grab the slider and tweak it down to lets say 120.

The difference to our LUT is this: First of all, the CDL workflow is only one operation in the end: Go from saturation 100 to saturation 120. With a LUT, we'd first add 50 to our saturation, and then set the saturation slider to 70 (subtracting 30 again) in order to have a nominal 120 – that's two counteracting operations and mathematically more destructive. Second – a LUT has a defined size and bit depth – the bigger both are, the better. However, the tools in nowadays grading softwares all work in floating point, which is much more accurate.

These differences are often not easily noticeable and in many scenarios not even considerable. However, the differences are there.



CDLs are all in all easier to transfer between different softwares and easier to maintain. They are also commonly used for exchange between vfx and editorial/grading/screening.

However, a LUT is the easier thing to apply and also makes sure the look stays defined and is not altered easily. Also, when creating a LUT, you do not have to stick to the CDL-parameters – you are welcome to use Lift, Gamma and Gain, or curves to achieve the look. So there's also a tad more creative freedom in the way you can create a certain look. This also accounts for technical LUTs, such as in color space conversion, or display calibration – in both of those scenarios, a CDL would prove impractical to use.

I hope this quick overview of LUTs and CDLs gave you enough information to decide for one or the other on your next project and also a few hints on what to look out for when using them.

Contributed by Mazze Anderhold, Assimilate.

In addition, the Director of Photography may have prepared a creative LUT, which adds a creative "look" to the footage. The DP may even have a different creative LUT for each scene. In most cases, both the technical and creative LUTs will be applied to the dailies so that the producers and editors can see an image that is as close to the DP's vision as possible.

Including these LUTs into the editor's dailies as well as the producers' dailies will save the editors a lot of time and headaches. While any editing software can apply a LUT to footage, it requires some extra organizational overhead, and it may slow down the editing machine a bit as well.

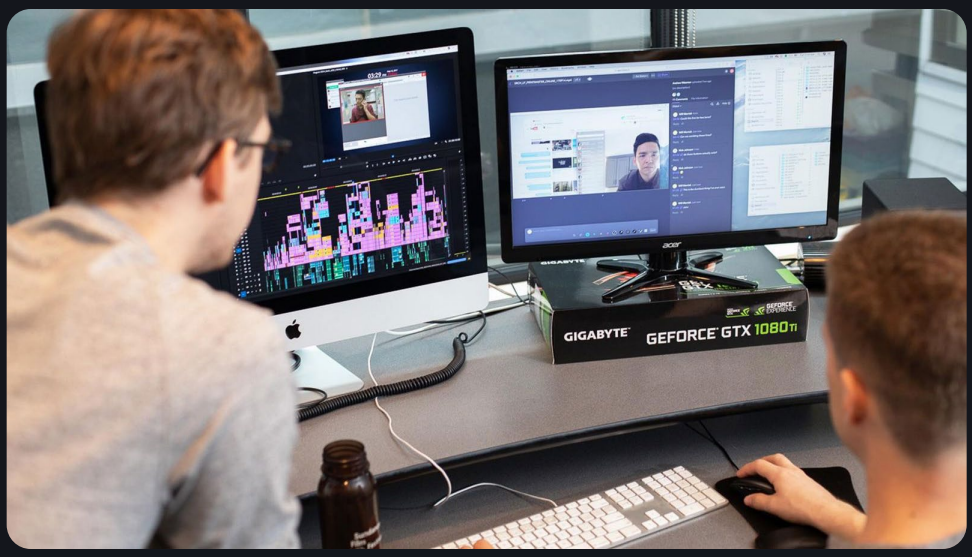
Later on, when the colorist is grading the footage, he or she will want to see the LUTs that the DP prepared. The colorist will be using the original camera files, not these transcoded dailies, but they will want to rebuild the look and feel of the footage as the DP saw it. Even if the colorist doesn't use those LUTs directly in the grading process, those LUTs provide a valuable reference for what the DP was thinking, and so it's important that all of the LUTs are passed through the pipeline.

On a smaller project, the DIT or a DIT's assistant may do most of this prep work. On feature films, it's common for all of the dailies work to be done by a separate post house, which then sends the files off to the editorial team. Depending on how complex the workflow is, how much footage is shot, and how large the files are, the process of creating dailies can take many hours, and so the dailies are sometimes more like yester-dailies.

## Files must be delivered to the editors and reviewers

On most projects, the dailies have to go out to two different groups: the editorial team, and the reviewers (producer, director,

client). Sending files to the editorial team is a fairly straightforward process. If the editorial team is nearby, it may be simplest to send a courier with a hard drive. If the editorial team is not nearby, files are often sent via a cloud service like Frame.io.




The duo editorial team on the indie hit "Searching" review producer and director comments in Frame.io.

Before digital cameras, all of the producers and the director would meet together in a theater to watch the dailies projected, since that was the only way to do it. That still does happen with digital dailies, but it's now much more common for the reviewers to watch the dailies separately online, using a platform like Frame.io which allows them to leave comments directly on the video from their laptop or smartphone.

The in-person scenario can allow for good discussions among the reviewers, but in many productions, the editorial team may not be present, and so those comments may not make their way into the editorial process. If the reviewers give feedback via Frame.io, all of their comments and ideas are attached to the clips themselves with timecode. That information can then be easily imported directly into the editing software for the editorial team to use when cutting the film.

In the feature film and television world, it is very common (and in most cases a mandatory aspect of the job) for the lead editor to be in working sessions with key reviewers (e.g. the director, DP, producer, etc.). Despite the power of remote editing processes available through a platform like Frame.io, in these larger production experiences, the in-person interaction is unavoidable.

But even in these scenarios, a remote workflow process is valuable for sharing dailies with studio execs, investors, or other third party stakeholders who might not sit in on early review sessions.

## 2 Real-world Examples of a Remote Dailies Workflow

Delivering dailies digitally can save time, money, and hard drive space. They also give you the added benefit of sharing dailies easily with your team, which lowers the barriers to creative collaboration.

Rather than hiring runners or a courier service, a properly designed digital dailies workflow can get your editor the files they need with a simple upload, even on a standard or unreliable internet connection.

So how do you design a workflow best suited for your project?

Let's first go over some of the major workflow components, then we'll dive into some real world examples to help show you how to put it all together.

But first, a quick word on how to think about your workflow.

## A Workflow is Always Evolving

The optimal workflow must consider all the different aspects and needs of a particular project at once. As you read through and digest this section, think of the different components of these workflows like a puzzle. Consider how changing one might affect the others.

One way to successfully approach a workflow is to always keep the end goals in mind. Is your top priority getting files to the editor as quickly as possible? Or is sharing synced dailies with a specific LUT applied to a high level producer your end game?

Just as important, make sure you've started an open dialogue with all of the post-production team members who will be affected by the workflow. You may not always think of a small change that has a big impact for the colorist if you're more of an editor.

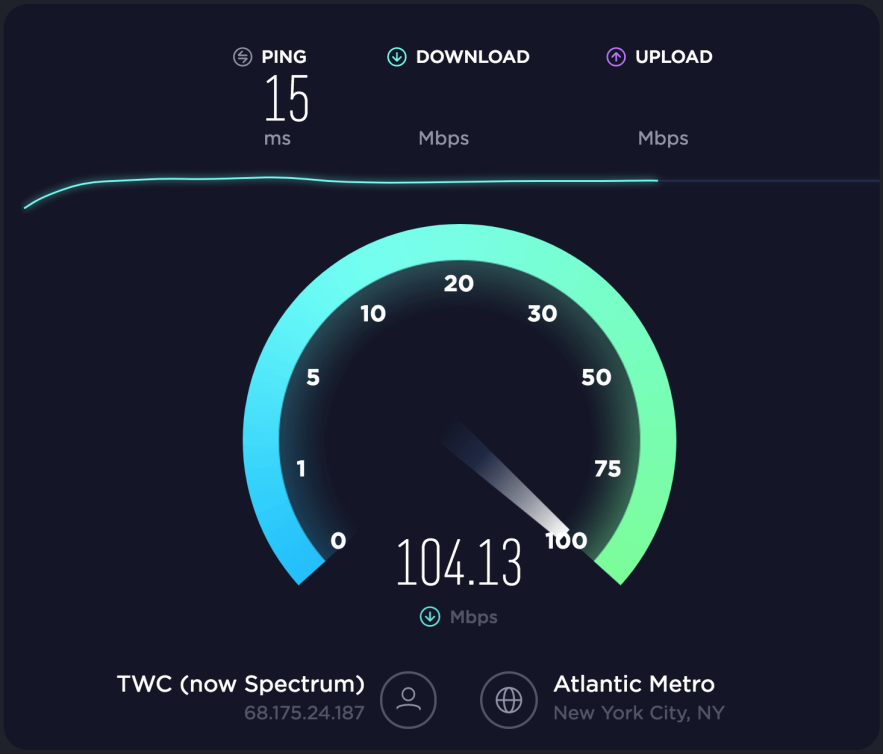
As you become more experienced and lock in tried-and-true methods, make sure to remain open to new ideas. Don't just recycle old habits because they're familiar.

Now, let's get started!

## Internet Connection

While it's really not necessary to have a lighting-fast internet connection to take advantage of uploading dailies (though it helps!), it's useful to know what you're working with.

Get an understanding of your upload and download speeds with a website like SpeedTest and then run a few real world tests of your own.




Upload some files to your chosen dailies server and time how long it takes. See what happens if you upload in batches versus all at once, or if the uploads stall out and have to be restarted with connection disruptions.

Give it as real a test as you can, either with similar footage from a past project, or by asking for some sample footage. Try to replicate a typical day's worth of files so you know exactly how long it'll take. The Simple Formula to Calculate Video Bitrates can help you calculate projected file sizes based on your dailies footage.

If you're on the production team, it's also a good idea to check in with your editor and make sure they've got a reliable connection. Downloading is typically much faster than uploading, so they'll have an advantage even with slower speeds, but loop them in on your test and make sure there are no issues on their end.

Once you have an idea of your upload speeds, you can begin to make adjustments to your dailies' resolution and codec to maximize your upload/download time.

## Camera, File Type, & Resolution (Frame Size)

The footage that you're starting with will determine a lot about your dailies options. If you're shooting in 4K, you'll likely want your dailies to fall somewhere in the 2K to HD range, unless your project is very short or your internet connection extremely fast.

Talk to your director of photography about the chosen camera's capabilities. Many high-end cameras have the ability to record proxy files alongside the raw footage and those proxies can be an excellent choice for dailies.

For example, RED cameras can record in any of the different ProRes formats. On RED projects, you can request 1920×1080 ProRes Proxy or ProRes422 as the in-camera proxy format, so that you can use these files for dailies.

This can be a huge time-saver, because it can allow you to skip the next step, transcoding dailies.

## Transcoding Dailies

If your camera doesn't natively support proxies, you can always transcode your RAW footage into an edit-friendly codec.

If you're on the production team, always discuss choosing a frame size and codec, with your editor. An editor may have a computer capable of editing 4K, but may prefer 2K files to save on disk space and get even better performance.

Or you might have an editor who prefers to be mobile, making HD dailies the right choice, both for greater performance on laptops and significantly less disk space requirements.

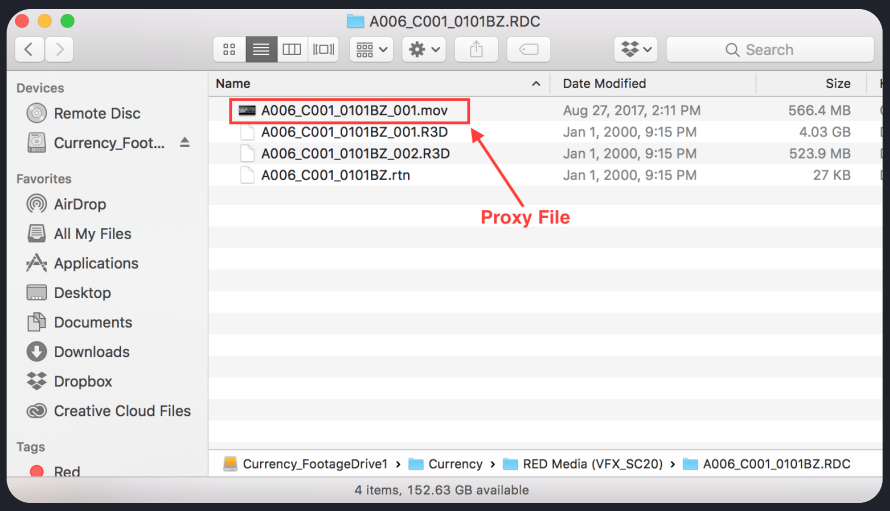
## To Sync or Not to Sync

If there's no sync sound in your project (usually commercials), then all you really have to worry about is having upload-friendly file sizes and an edit-friendly codec and resolution.

If your project has sync sound, consider again the purpose of your dailies upload. Are you simply delivering footage to the editor, or are you creating dailies that will be viewed by other team members, like a director or producer?

If the dailies are being reviewed by other team members, decide if they'll need synced footage. Often, a scratch audio track is recorded alongside the video and this will usually suffice for the purpose of reviewing dailies. However, in scenarios where there isn't a scratch track (such as if you're planning to sync by timecode alone) or the scratch track quality is very poor, you may decide to sync the dailies before passing them along to other team members.

If you're only delivering to the editor, then you have a bit more flexibility. For example, in the case of a RED shoot where ProRes proxies are being generated alongside the RAW footage and there's no need for transcoding, you may choose to begin uploading the dailies while simultaneously syncing and organizing the footage in your chosen NLE. At the end up each day, you would then upload the project file or bin for the editor, or back it up to a shared server.




If you don't have lower resolution edit-ready proxies coming directly out of the camera, then you'll likely choose to sync and transcode before uploading files for your editor.

Once you know the frame size and codec your editor prefers, you can use DaVinci Resolve to sync and transcode your dailies. Syncing is so easy in Resolve that in some scenarios (as illustrated below) it's worth re-exporting your footage even if you don't have to transcode for the edit.

## Real World Examples

Now that we've gone over the different aspects to consider when crafting a remote dailies workflow, let's take a look at how it all comes together with a few examples based on real-world projects.

### Ex #1: Feature Film, Sync in Premiere, Upload to Frame.io

Camera: RED Scarlet

Resolution: 4K

In-Camera Proxies: Yes

Shooting Location: Out of state

Assistant: Yes

This feature film was shooting several hours from the editing suite. That said, there was a very capable assistant editor on-set to prepare the footage, so there was no need for the main editor to be on set as well.

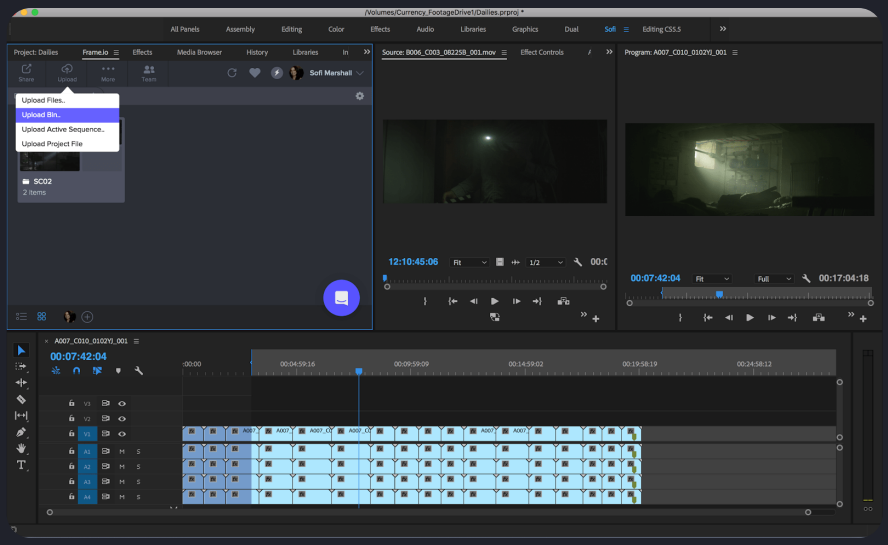
After talking with the DP and sound mixer (always do this

first!), this editorial team decided to use the Scarlet's in-camera proxy recording feature to remove the need for transcoding. 1920×1080 ProRes Proxy files were chosen, plus a reliable scratch audio track feeding into the camera.

The director had access to a dailies drive on set, but the team also wanted to be able to easily share the dailies with producers in LA, while controlling who could download and who couldn't. Frame.io allowed the team to deliver dailies to the editor and the producers in LA with just a single upload.

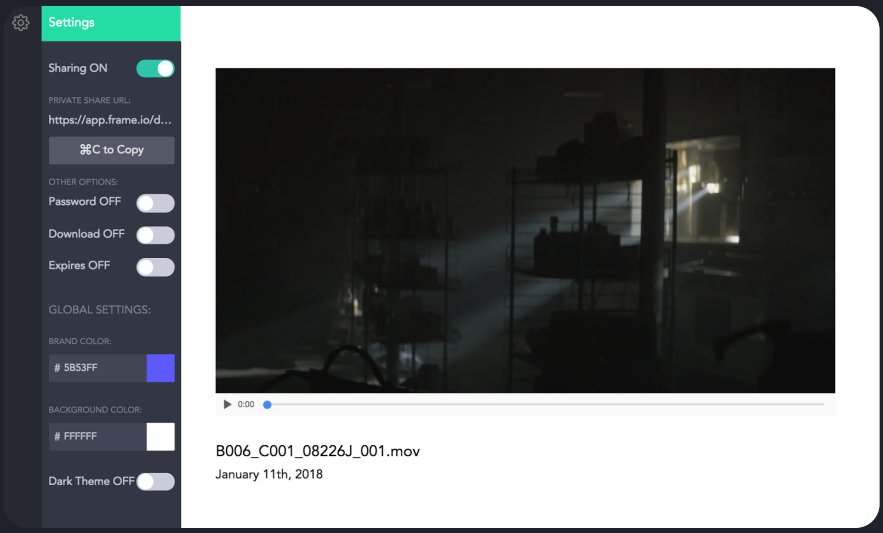
Because the location was very remote, the cast and crew were staying at a hotel with a pretty standard consumer level internet connection. Frame.io's robust upload management enabled them to deliver entire day's worth of footage, without much difficulty. The DIT would deliver the the in-camera proxies to the assistant editor, who then immediately ingested the footage into Premiere (ProRes Proxy files are edit-ready).

The assistance organized the footage by scene, and then used Frame.io's integrated Premiere panel to begin uploading the dailies.




While the dailies were uploading in the background, the assistant would sync the day's footage in Premiere, using

markers to add any comments or useful information from the editor reports right into the clips. Since the scratch track was reliable enough in this case, the unsynced footage was shared straight from the camera. This enabled the team to create custom dailies links within Frame.io to share with the producers in LA.




From those links, the producers could watch the dailies on whatever device they preferred—computer, phone or tablet—despite the files being somewhat large, without any buffering or stuttering problems. On top of that, this system allowed restrictions on downloading clips (when needed).

Finally, once the dailies uploaded, the editor was able to download them with the Premiere project, add each day's new bins to the master project, and maintain the notes that the assistant editor had added to the footage as markers right in the timeline.

This workflow worked very well. Despite a slower than ideal internet connection, the crew never had a problem leaving the dailies uploading in the background, and the producers in LA enjoyed being able to reliably access the dailies on the go.

Potential Adjustments

If this team needed to send the producers synced dailies (because the scratch track wasn't cutting it), that could be easily done. All you would need to do is auto-sync the in-camera proxies in Resolve, export out into the exact same codec and frame size (1920×1080 ProRes Proxy) and then go back to Premiere for organizing and uploading to Frame.io.

## Ex #2: Feature Film, Sync in Resolve, Upload to Frame.io

Camera: Canon C300

Resolution: 4K

In-Camera Proxies: Yes, with limitations

Shooting Location: Local

Assistant: No

As is the case for most independent films, pulling a PA away from set to run footage and drives back and forth is less than ideal. Additionally, the need to purchase additional shuttle drives and a large backup drive to store the raw files is a resource drain.

In this project, the sound recordist was capturing a reliable audio scratch track, so the unsynced dailies would be fine for the director or producers. On top of that, the DP chose the C300, partially for its capable in-camera proxy options.

However, the resolution and codec were predetermined and couldn't be changed without adjusting the RAW files' resolution.

| Video configuration of primary clips | | Video configuration of proxy clips[1] | | |
|---|---|---|---|---|
| Resolution | Frame rate | Resolution | Color sampling | Bit rate and compression[2] |
| 4096x2160<br>2048x1080 | 29.97P, 25.00P,<br>24.00P, 23.98P | 2048x1080 | YCbCr 4:2:0 8 bit | 24 Mbps, Long GOP |
| 3840x2160<br>1920x1080 | | 1920x1080 | | |
| 1920x1080 | 59.94i, 50.00i | 1920x1080 | | |
| 2048x1080 | 59.94P, 50.00P | 2048x1080 | | 35 Mbps, Long GOP |
| 1920x1080 | | 1920x1080 | | |

[1] The frame rate and audio recording settings will be the same as those used for primary clips.
[2] The camera uses a variable bit rate (VBR).

In this case, the proxies were set as 2K H.264 files wrapped in an .mfx, since the shooting resolution was 4K. Normally, H.264 files set off a red flag for most editors in terms of edit-friendliness, but the C300's proxies are incredibly tiny, totaling only about 10-12GB per full shoot day. The team wanted to take advantage of that.

After a few tests, it was determined that playback wasn't a deal-breaker, but it wasn't perfect either. These files definitely stuttered, and there was delay when scrubbing clips in the viewer. But, much of the editing was going to be done while in the field/travelling, so the team wanted files that were small enough to fit on a portable, bus-powered hard drive, but still perform well on a laptop.

The team's tests showed that the only way to make editing speed on the laptops decent was to decrease playback resolution significantly. Another piece of the puzzle to consider was that there was no assistant editor due to budgetary restrictions (read: independent film). Since all syncing was being handled by the editor, the team wanted it to happen as quickly and painlessly as possible.

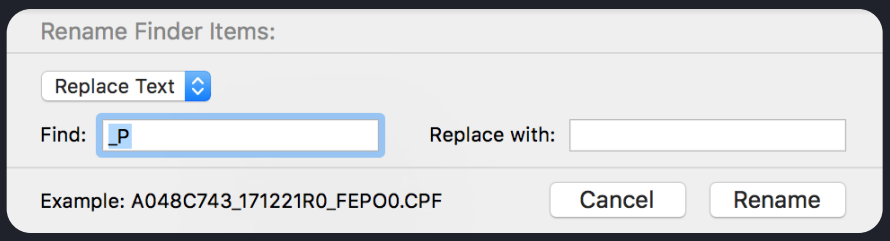
So what did they do?

As was alluded to before, they turned to Resolve for batch syncing. As the team learned, as long as the files do not change resolution, Resolve can transcode from H.264 to ProRes Proxy very quickly.

The team's tests proved that their ultraportable laptops could handle 2K ProRes files without much trouble. So, they decided to upload the C300's tiny, in-camera dailies after each shoot day, bring them into Resolve for audio sync, and then transcode to 2K ProRes Proxy dailies to use in the edit.

It took about 30 minutes each day to sync and transcode a day's worth of footage with the editor's laptop. While this process added an extra step in the dailies workflow, it was completely viable with the available hardware, and saved a lot of time in the long run (because of the improved editing performance).

[Note: The C300 has a unique feature that adds a "_P" to the end of its proxy clips. That should be a big red flag for you if you've ever done an online edit, as you generally want the proxy and master clips to have the exact same file names. To eliminate this issue, simply select the entire contents of each day's proxy folder, right click and select rename. In the rename dialogue, use replace text to find the _P and replace it with nothing.)




Again, Frame.io is a good choice for this example workflow, as it allows a director to have easy access to the in-camera proxies even as they travel, or during a break on set. The editor could have easily shared the proxy files after sync with only a few clicks.

Potential Adjustments

If you don't travel during a project, or are working on a high-

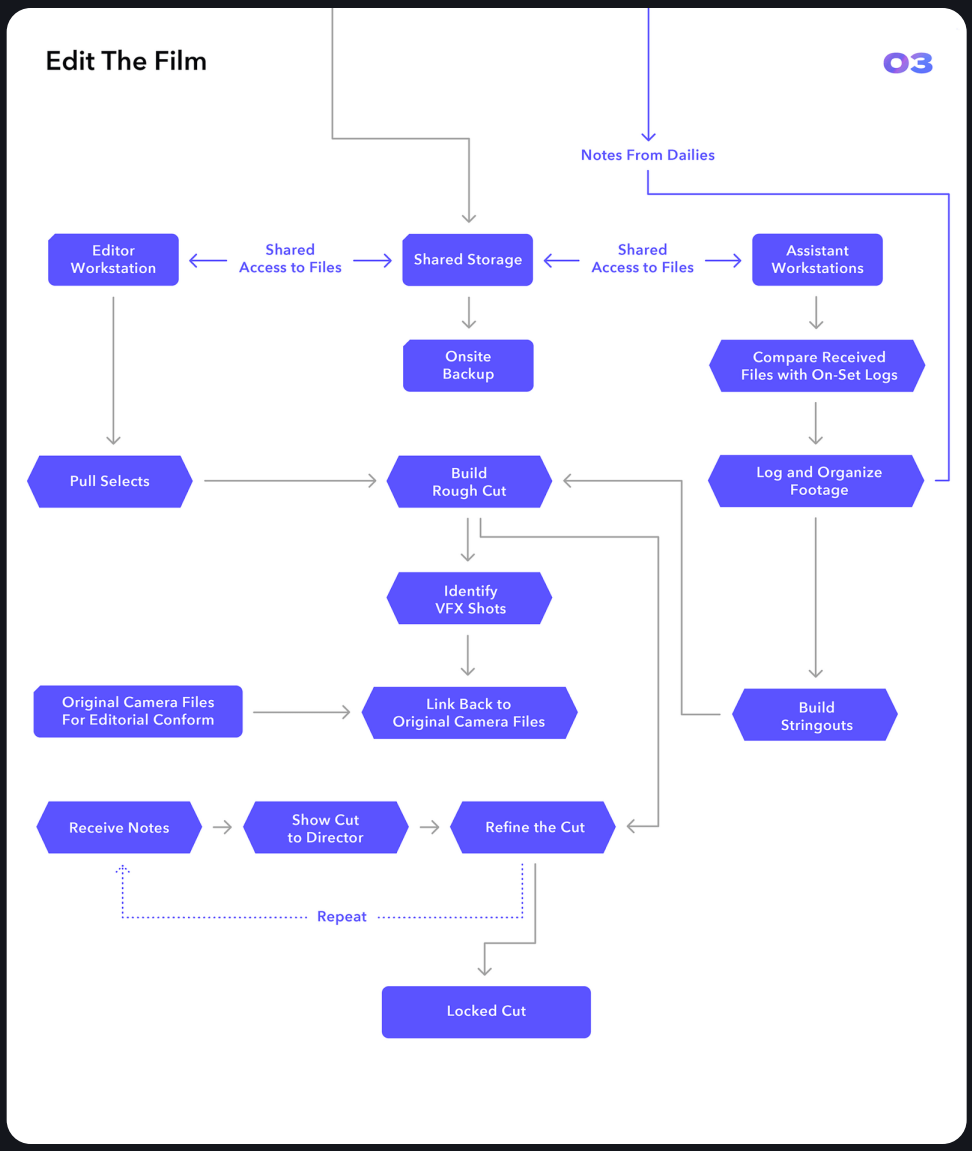
powered desktop, then there's a good chance the playback of the original in-camera proxies would be completely fine for the edit. In that case, you may not need to transcode in Resolve. However, in that case you would need to sync clips individually in an NLE, or would have to turn to something like PluralEyes.

CHAPTER 03

# Edit

Whether it takes a team of twenty or a solo artist, the editorial process is balance of artistry and technical details.

*16 minute read*




As soon as the dailies arrive on a hard drive (or over the internet), the assistant editors do their own preparation, known as *ingest*. After ensuring that all of the files have arrived as expected and that the sound and metadata are properly synced, the first step is to copy the dailies to their central shared storage.

## Shared Storage

Unless the editorial team is just one person, you will need storage that can be shared by multiple editors. The assistants and the lead editor will be working in tandem throughout the project, and they all need access to the same files.

The editorial team

The size of the team varies greatly based on the project in question. A 30-second advertisement will seldom have more than two. A wedding videography business may have just one person doing all of the editing-related tasks, and a large feature film may have half a dozen.

Even on Hollywood films, the size of editorial teams can vary widely from just 2 people to 8 or 10 people on an extremely complex film.

On almost all high-end work, however, there is an editor and at least one assistant.

The editor is the primary person responsible for making the creative decisions about which shots and takes to use, when to cut, etc. On some larger films, it's becoming more common to have two people share the title of editor, especially when the film is working on a tight schedule.

In order to allow the editor(s) to focus on the story, the assistant editors keep everything organized and in sync. The assistants are the workflow masterminds and must have much deeper technical savvy than the lead editor. Depending on the size of the project, the amount of footage, and the amount of VFX, there may also be a VFX editor.

Contributed by Lisa McNamara, Freelance writer and editor.

While it's possible for each member of the editorial team to keep their own copy on their own external hard drive, it's a nightmare to keep everyone's changes in sync.

Shared storage solves this problem by allowing everyone on the team to connect to a single data server which holds all of their files in one place. The simplest shared storage, a box the size of a small desktop computer with 4 hard drives inside it, might cost only $400. On the high end, a large post house may have dozens of rack-mounted servers that cost hundreds of thousands of dollars.

Jellyfish shared storage rack array. Image courtesy of LumaForge.

The primary factors are speed (how fast can you copy files in and out?), simultaneous connections (how many people can access the files at once?), storage size (how many GB or TB can you store?), and reliability (what happens if a hard drive crashes?).

Deep Dive: Shared Storage

The topic of shared storage is built on a foundation of two ideas: storage and sharing (or technically: networking) which we can cover separately before we bring them together.

Storage is the concept of storing all of the files for a particular project or workflow. They may not all be stored in the same place, because different types of data have different

requirements, different storage solutions have different strengths and features.

At a fundamental level, most digital data is stored on HDDs or SSDs. HDDs or Hard disk drives are mechanical devices that store the data on a spinning magnetic surface, and move read write heads over that surface to access the data. They currently max out around 200MB/s and 5ms latency. SSD stands for solid state drive, and involves no moving parts. SSDs can be built with a number of different architectures and interfaces, but most are based on the same basic flash memory technology as the CF or SD card in your camera. Some SSDs are SATA drives that use same interface and form factor as a spinning disk, for easy replacement in existing HDD compatible devices. These devices are limited to SATA's bandwidth of 600MB/s. Other SSDs use the PCIe interface, either in full sized PCIe cards, or the smaller M.2 form factor. These have much higher potential bandwidths, up to 3000MB/s. Currently HDDs are much cheaper for storing large quantities of data, but require some level of redundancy for security. SSDs are also capable of failure, but it is a much rarer occurrence. Data recovery for either is very expensive. SSDs are usually cheaper for achieving high bandwidth, unless large capacities are also needed.

## RAID

Traditionally hard drives used in professional contexts are grouped together for higher speeds and better data security. These are called RAIDs, which stands for Redundant Array of Independent Disks. There are a variety of different approaches to RAID design, which are very different from one another.

RAID 0 or striping is technically not redundant, but every file is split across each disk, so each disk only has to retrieve its portion of a requested file. Since these happen in parallel, the

result is usually faster than if a single disk had read the entire file, especially for larger files. But if one disk fails, every one of your fails will be missing a part of it, making them all pretty useless. The more disks in the array, the higher the chances of one failing, so I rarely see striped arrays composed of more than 4 disks. It used to be popular to create striped arrays for high speed access to restorable data, like backed up source footage, or temp files, but now a single PCIe SSD is far faster, cheaper, smaller, and more efficient in most cases.

RAID 1 or mirroring is when all of the data is written to more than one drive. This limits the array's capacity to the size of the smallest source volume, but the data is very secure. There is no speed benefit to writes since each drive must write all of the data, but reads can be distributed across the identical drives with similar performance as RAID 0.

RAID 4,5 & 6 try to achieve a balance between those benefits, for larger arrays with more disks (Minimum 3). They all require more complicated controllers, so they are more expensive to reach the same levels of performance. RAID4 stripes data across all but one drive, and then calculates and stores parity (odd/even) data across the data drives and stores it on the last drive. This allows the data from any single failed drive to be restored, based on the parity data. RAID5 is similar, but the parity volume is alternated depending on the block, allowing the reads to be shared across all disks, not just the "data drives." So the capacity of a RAID 4 or 5 array will be the minimum individual disk capacity, times the number of disks minus one. RAID 6 is similar, but stores two drives worth of parity data, which via some more advanced math than odd/even, allows it to restore the data if even two drives fail at the same time. RAID 6 capacity will be the minimum individual disk capacity, times the number of disks minus two, and is usually only used on arrays with many (>8) disks. RAID5 is the most popular option for most media storage arrays, although RAID6 becomes more popular as the value of the data stored increases and the price of extra drives decreases

over time.

## Connection Speed

Measuring Data:

Digital data is stored as a series of ones and zeros, each of which is a bit. 8 bits is one byte, which frequently represents one letter of text, or one pixel of an image (8-bit single channel). Bits are frequently referenced in large quantities to measure data rates, while bytes are usually referenced when measuring stored data. I prefer to use Bytes for both purposes, but it is important to know the difference. A Megabit(Mb) is one million bits, while a Megabyte (MB) is one million bytes, or 8 Million bits. Similar to metric, Kilo is thousand, Mega is million, Giga is billion, and Tera is trillion. Anything beyond that you can learn as you go.

Storage Math:

Technically a Kilobyte is 1024 bytes (2^10) according to JEDEC who determines the standards for memory, but is 1000 (10^3) according to most storage manufacturers. This (as well as formatting losses) is why the OS usually reports a smaller storage volume than is written on the side of the device. More maximum precision kilobytes counted as 1024 bytes are supposed to be abbreviated KiB, and Megabytes counted as 2^20 or 1,048,576 bytes should be labeled as MiB, but this standard is not strictly followed. Most calculations are made based on 1000 for simplicity, but difference is 10% by the time you get to Terabyte in the Trillions, so it is best to account for that when estimating storage needs. But you should be allowing for more than 10% head room anyway in most cases.

Networking speeds are measured in bits (Gigabit) but with headers and everything else, it is safer to divide by ten when converting speed into Bytes per second. Estimate 100MB/s for Gigabit, up to 1000MB/s on 10GB, and around 500MB/s for the new N-BaseT standard. Similarly when transferring files over a 30Mb internet connection, expect around 3MB/s. Then multiple by 60 or 3600 to get to minutes or hours. (180MB/min or 9600MB/hr in this case.) So if you have to download a 10Gigabyte file on that connection, come back to check on it in an hour.

## Top End Networking Options

40Gb Ethernet products have been available for a while, and we are now seeing 25GB and 100Gb Ethernet products as well. 40Gb cards can be gotten quite cheap, and I was tempted to use them for direct connect, hoping to see 4GB/s to share fast SSDs between systems. But 40Gb Ethernet is actually a trunk of four parallel 10Gb links, and each individual connection is limited to 10Gb. It is easy to share the 40Gb of aggregate bandwidth across 10 systems accessing a 40Gb storage host, but very challenging to get more than 10Gb to a single client system. Having extra lanes on the highway doesn't get you to work any faster if there are no other cars on the road, it only helps when there is lots of competing traffic. 25Gb Ethernet on the other hand will give you access to nearly 3GB/s for single connections, but as that is newer technology, the prices haven't come down yet. ($500 instead of $50 for a 10GbE direct link.) 100Gb Ethernet is four 25Gb links trunked together, and subject to the same aggregate limitations as 40Gb.

Because networking standards are measured in bits, and networking is so important for sharing video files, many video file types are measured in bits as well. An 8Mb H.264 stream is 1MB per second. DNxHD36 is 36Mb/s (or 4.5MB/s when

divided by 8) DV and HDV are 25Mb, DVCProHD is 100Mb, etc. Other compression types have variable bit rates depending on the content, but there still average rates we can make calculations from. Any file's size divided by its duration will reveal its average data rate. It is important to make sure that your storage has the bandwidth to handle as many streams of video as you need, which will be that average data rate times the number of streams. So ten streams of DNxHD36 will be 360Mb or 45MB/s.

## IOs and Latency

Lots of small requests require not just high total transfer rates, but high IO performance as well. Hard drives can only fulfill around 100 individual requests per second, regardless of how big those requests are. So while a single drive can easily sustain a 45MB/s stream, satisfying 10 different sets of requests may keep it so busy bouncing between the demands that it can't keep up. You may need a larger array, with a higher number of (potentially) smaller disks to keep up with the IO demands of multiple streams of data. Audio is worse in this regard, in that you are dealing with lots of smaller individual files as your track count increases, even though the data rate is relatively low. SSDs are much better it handling larger numbers of individual requests, usually measured in the thousands or tens of thousands per second per drive.

Capacity on the other hand is simpler. Megabytes are usually the smallest increments of data we have to worry about calculating. A media type's data rate (in MB/sec) times its duration (in seconds) will give you its expected file size. If you are planning to edit a feature film with 100 hours of offline content in DNxHD36, that is 3600×100 seconds, times 4.5MB/s, is 1620000MB, or 1620GB, or simply about 1.6TB. But you should add some head room for unexpected needs, and then a 2TB disk is about 1.8TB when formatted, so it will

just barely fit. But it is probably worth sizing up to at least 3TB if you are planning to store your renders and exports on there as well.

Once you have a storage solution of the required capacity, there is still the issue of connecting it to your system. The most expensive options connect through the network to make them easier to share (although more is required for true shared storage) but that isn't actually the fastest option, nor the cheapest. A large array can be connected over USB3 or Thunderbolt, or via the SATA or SAS protocol directly to an internal controller. There are also options for Fiber Channel, which can allow sharing over a SAN, but this is becoming less popular as 10GbE becomes more affordable. Gigabit Ethernet and USB3 won't be fast enough for high bandwidth files to playback, but 10GbE, multichannel SAS, Fiber Channel, and Thunderbolt can all handle almost anything up to uncompressed 4K. Direct attached storage will always have the highest bandwidth and lowest latency, as it has the fewest steps between the stored files and the user. Host->SAS-RAID-Controller->SAS-Drives. Using Thunderbolt or USB adds another controller and hop, Ethernet even more so.

## Types of Project Data that need to be Stored

Now that we know the options for storage, let's look at the data we anticipate needing to store. First off we will have lots of video footage of source media. (Either camera original files, transcoded editing dailies, or both.) This is usually in the Terabytes, but the data rates vary dramatically, from 1Mb H.264 files, to 200Mb ProRes files, to 2400Mb Red files. The data rate for the files you are playing back, combined with the number of playback streams you expect to use, will determine the bandwidth you need from your storage system. These files are usually static, in that they don't get edited or written to in any way after creation. The exceptions would be sidecar files

like RMD and XML files, which will require write access to the media volume. If a certain set of files are static, as long as a backup of the source data exists, they don't need to be backed up on a regular basis, and don't even necessarily need redundancy. (Although if the cost of restoring that data would be high, in regards to lost time during that process, some level of redundancy is still recommended.)

Another important set of files we will have is our project files, which actually record the "work" we do in our application. They contain instructions for manipulating our media files during playback or export. They files are usually relatively small, and are constantly changing as we use them. That means they need to be backed up on a regular basis. The more frequent the backups, the less work you lose when something goes wrong.

We will also have a variety of exports and intermediate renders over the course of the project. Whether they are flattened exports for upload and review, or VFX files, or other renders, these are a more dynamic set of files than our original source footage, and are generated on our systems, instead of being imported from somewhere else. These can usually be regenerated from their source projects if necessary, but the time and effort required usually makes it worth it to invest in protecting or backing them up. In most workflows, these files don't change once they are created, which makes it easier to back them up if desired.

There will also be a variety of temp files generated by most editing or VFX programs. Some of these files need high speed access for best application performance, but they rarely need to be protected or backed up, because they can be automatically regenerated by the source applications on the fly if needed.

## Where to store everything

So we have source footage, project files, exports, and temp files that we need to find a place for. If you have a system or laptop with a single data volume, the answer is simple, it all goes on the C-Drive. But we can achieve far better performance is we have the storage infrastructure to break those files up onto different devices. Newer laptops frequently have both a small SSD, and a larger hard disk. It that case we would want our source footage on the (larger) HDD, while the project files should go on the (safer) SSD. Usually your temp file directories should be located on the SSD as well since it is faster, and your exports can go either place, preferably the SSD if they fit. If we have an external drive of source footage connected, we can back all files up there, but should probably work from projects stored on our local system, playing back media from the external drive.

Now a professional workstation can have a variety of different storage options available. I have a system with two SSDs and two RAIDs, so I store my OS and software on one SSD, my projects and temp files on the other SSD, my source footage on one RAID, and my exports on the other. I also back up my project folder to the exports RAID on a daily basis, since the SSDs have no redundancy.

## Individual Store Solution Case Study Examples

If you have a short film project shot on Red, that you are editing natively, R3Ds can be 300MB/s. That is 1080GB/Hour, so 5 hours of footage will be just over 5TB. It could be stored on a single 6TB external drive, but that won't give you the bandwidth to play back in real time. (Hard drives usually top out around 200MB/s) Now striping your data across two drives, in one of those larger external drives would probably provide the needed performance, but with that much data

you are unlikely to have a backup elsewhere. So data security becomes more of a concern, leading us towards a RAID5 based solution. A 4 disk array of 2TB drives provides 6TB of storage at RAID5. (2TB*[4-1]) This will be more like 5.5TB once it is formatted, but that might be enough. Using and array of 8 1TB drives would provide higher performance, and 7TB of space before formatting (1TB*[8-1]) but will cost more. (8 port RAID controller, 8 bay enclosure, and two 1TB drives are usually more expensive than one 2TB drive.)

Larger projects deal with much higher numbers. Another project has 200TB of RED footage that needs to be accessible on a single volume. A 24Bay enclosure with 12TB drives provides 288TB of space, minus two drives worth of data for RAID6 redundancy (288TB-[2x12TB]=264TB) which will be more like 240TB available in Windows once it is formatted.

## Sharing Storage and Files with Others

Initially, most shared storage solutions were developed based on the fiber channel interface. A fiber channel array appears as a local drive to a host system connected to it with a fiber channel interface card. And it is possible to direct connect storage with Fiber channel, allowing the loud array to be safely stored away far from the workstation, at no loss in performance. But the real power of Fiber Channel over SCSI or SAS comes when you add a switch into the equation. Now many systems can access the same array(s). They can either all access separate dedicated volumes, or access the same volumes and data, as long as they have a system to prevent them from overwriting and corrupting each other's data.

## SANs

"Shared storage" technically has two very different meanings. In a data center, servers need high speed, high IO storage available, and before SSDs, this meant lots of individual drives. Instead of every server having their own array, many servers would share a single array of disks, each having their own dedicated portion of the entire capacity. Each server had dedicated access to their files, and no other server could see them, unless they accessed them over the network through the host machine, the same as if the files were stored on an array inside that server. The advantage were that fewer arrays were needed to provide the performance that various servers needed at their peak, and storage volume could be adjusted to meet the changing needs of various servers without having to move physical hard drives around. They were just given a larger portion of the array if they started to fill their allotment, and eventually more arrays were added. This sharing of the physical storage, but not the files contained on them, was usually accomplished over the Fiber Channel interface, in what was referred to as a Storage Area Network, or SAN. It is now also done over Ethernet networks, usually in the form of iSCSI.

As technology advanced, it became obvious that allowing more than one machine to access the files on a large array could be advantageous, as multiple servers could host the same web site, improving performance by sharing the processing load. For media production, that meant allowing multiple workstations to see the same source footage, at full local storage speeds. This was easy for read operations, requests get sent to the drive, and it fulfills them as fast as it can. But for write operations, how do we keep multiple machines from writing to the same spot on the drive at the same time? Either one machine controls the drive and all write operations have to be approved by it, slowing down the process, or control is passed to whichever machine needs it, which can be complicated to manage. So true shared SAN software is usually expensive, on top of the price of the arrays you are sharing and the infrastructure to share it.

Originally Shared SANs exclusively ran on expensive dedicated fiber optic networks using an interface called fiber channel, separate from the Ethernet network that machines used to communicate with one another. Eventually there was a move to FCoE or Fiber Channel over Ethernet, and then to iSCSI, which was entirely Ethernet packet based, lowering the cost of the hardware. But the software challenge remained the same, to prevent multiple machines who all see the drive as a local volume for writing to the same spot at the same time. FiberJet, MetaSAN, StorNex, XSAN, Facilis, Harmony, Avid Unity, and many others all had different approaches to doing that.

As Ethernet networking technology improved, the benefits of SAN solutions over NAS (Network attached storage) solution diminished. 10Gigabit Ethernet (10GbE) transfers over 1GB of data a second, and is relatively cheap to implement. NAS has the benefit of a single host system controlling the writes, usually with software included in the OS, preventing data corruption, and also isolated the client devices from the file system, allowing PC, Mac, and Linux devices to all access the same files. This came at the cost of slightly increased latency, and occasionally lower total bandwidth. But the prices and complexity of installation were far lower. So now all but the largest facilities and most demanding workflows are being deployed with NAS based shared storage solutions. This can be as simple as a main editing system with a large direct attached array sharing its media with an assistant station, over a direct 10GbE link, for about $50. This can be scaled up by adding a switch and connecting more users to it, but the more users sharing the data, the greater the impact on the host system, and the lower the overall performance. Over 3-4 users, it becomes prudent to have a dedicated host system for the storage, for both performance and stability. Once you are buying a dedicated system, there are a variety of other functionalities offered by different vendors to improve performance and collaboration.

There is one other solution on the market for sharing media at high speeds, and that is Thunderbolt3. There are vendors that sell products that allow direct connections to systems via Thunderbolt3, but most of those are using the inherent support in Thunderbolt for TCP networking to move the data, effectively making it a NAS on the Thunderbolt interface. Any product that did offer block level access over Thunderbolt3, appearing as a local Thunderbolt drive to the workstation, would still need the same SAN software functionality required by Fiber Channel to prevent overwrites and corruption. The advantage of Thunderbolt over Fiber Channel would be price and speed, at the expense of distance and scalability.

## Bin Locking

The main step to improve collaboration is to implement what is usually referred to as a "bin locking system." Even with a top end SAN solution, and strict permissions controls, there is still the possibility of users over-writing each other's work, or at the very least branching the project into two versions that can't easily be reconciled. If two people are working on the exact same sequence at the same time, only one of their sets of changes is going to make it to the master copy of the file, without some way of combining the changes. (And solutions are being developed) But usually the strategy to avoid that is to break projects down into smaller pieces, and make sure that no two people are ever working on the exact same part. This is accomplished by locking the part (or bin) of the project that a user is editing, so that no one else may edit it at the same time. This usually requires some level of server functionality, as it involves changes that are not always happening at the local machine. Avid requires specific support for that from the storage host for it to enable that feature. Adobe on the other hand has implemented a simpler storage based solution, which is effective but not infallible, that works on any shared storage device that offers users write

access. A lock file next to the project file indicates to other users who is using that file, and prevents their applications from attempting to alter the project. This prevents branches and overwriting if users follow the rules. Avid's solution is stronger and more fool-proof, but requires explicit support from the storage host. There are a variety of vendors besides Avid who offer this functionality.

## Shared Storage Case Studies

If I was setting up a facility with 10 editors in the current environment, I would base it on 10GbE for sure, and have a dedicated system hosting the storage, with a large RAID for the media and exports, and smaller SSD volume for the project files. I would have an SSD on every system for the applications and temp files, to save some bandwidth to the storage server.

iSCSI arrays offer some interesting possibilities for read only data, like source footage, as iSCSI gives block level access for maximum performance, and runs on any network without and expensive software. The only limit is that only one system can copy new media to the volume, and there must be a secure way to ensure the remaining systems have read-only access. Projects and exports must be stored elsewhere, but require much less capacity and bandwidth than source media.

Contributed by Mike McCarthy, Tech with Mike First.

## RAIDs

Video files keep getting bigger. Every year we get more pixels, more colors, and more data. Thankfully, computer

storage has generally kept pace, with larger and larger drives available at lower and lower prices. However, in the era of digital video workflows the problem is not just where to fit all our footage, but how to work with it efficiently. This is where RAID comes in handy.

RAID stands for *Redundant Array of Independent Disks*. As you can probably guess from the name, RAIDs are a method for connecting multiple physical drives to act as a single virtual storage device. This lets the drives work together to increase performance, add redundancy, or a mix of both. RAID is not the only way to use a bunch of drives together, but it is the most common.

## RAID vs Single Drives

Given the choice of a single large hard drive or a RAID of the same capacity, you should almost always choose the RAID. Why? Because RAIDs far exceed the capabilities of a single drive. Most individual hard drives are just too slow for smooth editing of 4K+ files, but multiple hard drives in a RAID can split up the work to deliver much faster read/write speeds. Not only that, but many RAIDs offer at least some measure of fault tolerance (the ability to survive drive failure). If you have one drive and it fails, that's it, your data is lost. But certain RAID configurations can have up to half of their drives fail without you losing anything. That's a big deal.

The only aspect of RAID that is worse than using a single drive is the cost. If you need 8TB of storage, an 8TB RAID will be more expensive than an 8TB hard drive. That said, you're not really getting the same thing from both these options. Yes, both options will store the same amount of data, but what you're really paying for is the added benefits of RAID. If you wanted to buy a single drive that was *as fast and high capacity* as a RAID, it would be much more expensive (if you could find

one at all). And of course, there is no way for a single disk to be redundant by itself, so if that's what you need then RAID is the better (only) option. In general, RAIDs deliver the best performance and highest redundancy for the least cost.

RAIDs come in a wide variety of (physical) shapes and (storage) sizes, so you can add the benefits of RAID to practically any workflow. Headed into the field for a remote editing job and only need a couple terabytes just for yourself? Take a small mobile RAID along to edit faster or with a little more redundancy (the smallest RAIDs can't offer both benefits at once, but that's still better than a typical external drive). Running a studio with a dozen editors and several hundred terabytes of footage? Configure a rack-mounted RAID server to meet your team's particular technical requirements.

## How RAID Works

Different configurations of RAID, called *levels*, are each optimized for certain attributes that trade between cost, redundancy, and speed in a particular way. Even though RAIDs all do the same basic thing (store files), there are distinct advantages and disadvantages to each level. So, you will need to pick RAID options carefully for each task in your workflow. But before you can decide on the best options for your production, you first need to understand the basics of how RAID works.

All RAIDs are made up of roughly the same parts; multiple drives (from two to hundreds) inside an enclosure of some kind, and a controller. Controllers instruct the array of drives how to handle data, so in essence, the controller is what determines what level the RAID will be. The controller can be purpose-built hardware that actually attaches to the drives, or it can be software on the computer connected to the array. There are three fundamental techniques that controllers use

to handle the data on the RAID; data striping, data mirroring, and data parity.

*Data striping* is a technique for cutting up sequential data and storing the cut-up pieces to separate drives. This allows for faster performance, because the multiple drives work together to read/write the file, so the task is completed in less time. For example, a RAID can split a 1GB video into smaller pieces, and write those pieces to separate drives. When the file needs to be accessed again, each drive only has to retrieve their portion of the data for the controller. That means you can get at least double the performance of a single drive from a striped array. However, as appealing as this increased speed is, data striping also brings risk. If any of the file's data is lost on one drive, then the entire file is lost. Even if there are 10 drives, with data striped equally across them, one drive failure will cause total data loss. The files simply can't be put back together if there is a missing piece, no matter how small. This is a considerable problem, because the more drives in an array, the greater the chance of a drive failure occurring.

This risk is why some RAID levels also employ *data mirroring*. Data mirroring is a technique that stores a file (or piece of a file) identically across multiple drives. This does not increase the speed of writing data, but it does provide huge gains in redundancy. If that same 1GB video file was stored on a RAID that used data mirroring across two disks, then one of those drives could fail completely without any data being lost, because the full file is stored on both drives. And even though file write times are not improved, file read times are faster, since the controller can still read a file from separate drives at the same time. The biggest downside of data mirroring is that it does not increase the total usable storage of the array. 2 1TB drives mirrored to each other will create a RAID with a capacity of only 1TB. That means this array is 2 times the cost of a single 1TB drive, but has the same capacity. Redundancy is useful because it reduces risk, but in this regard it's also expensive.

*Data parity* is a way to allow you to keep an internal backup without requiring double the number of drives. Using some data processing magic, parity calculations allow you to use one drive in a RAID array to back up multiple other drives at the same time. Parity data is not a full backup, however. On its own, parity data is meaningless, but it's possible to reconstruct a lost drive by combining the parity data with the other drives in the array. Parity adds an extra level of complexity, requires a higher-performance controller, and increases drive fatigue, which introduces new risks that have to be accounted for by the user.

If you think this is all too complicated for your workflow, don't worry. You don't need to be a computer engineer or programmer to use RAID. The controller does most of the hard work managing the data, so you can relax. That said, you should decide whether hardware or software-based RAID fits your needs best. Hardware RAID controllers are generally more expensive, but the performance is often better. You also won't need to install any special software on your workstation. That means you can plug the RAID into basically any machine, and it will just work. Software RAID, on the other hand, is more affordable, and still provides great performance. However, because the controller is software, you can change the RAID configuration as you need from project to project. Not all hardware controllers have this same level of flexibility. That said, these arrays are not as easy to set up. Most RAID levels are possible with either hardware or software controllers, it just depends on your budget and technical requirements.

Now that you know what RAIDs do with your data, you can pick RAID solutions that meet your needs. You don't even need to start from square one to decide. Several RAID levels are already proven for certain parts of post-production workflows.

## Standard RAID Levels

RAID 0 is an array of drives with striped data. This offers wonderful speed and simplicity, as using only two drives can double your read and write speeds (compared to a single drive). However, as mentioned before, using multiple drives to stripe data also increases the chance of data loss. If either drive fails, then you lose all of your data. A good use case for RAID 0 is as an editing drive or scratch disk, where maximum performance is necessary, but the media is backed up somewhere else. RAID 0 is especially useful for mobile editing solutions, because you only need a small, portable array of two disks. That said, only use this configuration if you make frequent backups of the data you're putting on the array.

RAID 1 is an array of mirrored drives. This delivers excellent redundancy, while also being easy to maintain. You only get write performance equal to a single drive, but you get double read speeds, because the array still reads files from multiple disks at once. RAID 1 only needs a minimum of 2 drives, so these units can also be quite portable, making them very useful for camera offloads and field backups. The biggest downside is the capacity and performance you get for the cost. In order to mirror data, the total array capacity can only be as big as the smallest drive. So, for a 2-drive RAID 1 you've paid for two drives, but can only use the capacity of one.

RAID 10 (1+0) is a combination of RAID 1 and RAID 0, where half the drives are mirrored to each other, and then data is striped between the mirrored pairs. This gives you good read/write performance, while also allowing for a significant level of redundancy. In the best case scenario, up to half the drives in a RAID 10 can fail without any data loss, but you still get double performance of any single drive. RAID 10 is very useful for many different applications. Just like the lower levels of RAID, it can be used as an editing drive for one user. But where RAID 10 really shines is in input-output intensive tasks, like media servers and databases for multiple simultaneous

users. The biggest downside of this configuration is that you only get half the total capacity of the arrayed drives. RAID 10 requires more drives from the start (at least 4), so you may not like paying for storage capacity that you cannot use. Overall, RAID 10 gives you a fantastic balance of performance and redundancy. It is an excellent solution for many workflow tasks that need high data throughput.

Pretty much any post-production team can benefit from these RAID levels. They are easy to understand, simple to use, and widely available from a large number of vendors. However, beyond a certain scale, these levels can incur unnecessary cost. 100TB in RAID 10 would require 200TB of drives. Do you really want to pay for all that? It would certainly be a robust system, but that's money you could be using elsewhere in your project.

## Advanced RAID Levels

The good news is that there are other RAID levels that let you to use more of the total storage capacity within an array. The bad news is that these RAID levels are more complicated, and introduce new risks that should be carefully considered before you use them in your workflow.

RAID 5 is a striped array with distributed parity. That means that every time a file is written to this kind of array it is striped across the drives, and then a parity calculation is performed, which is also written to the array. Parity files are not all kept on the same drive, but are distributed equally across them. Striping gives RAID 5 improved performance, while distributed parity allows it to survive the failure of a single drive. But the main advantage of RAID 5 is that it gives you more storage for your money. If you had 5 8TB hard drives, RAID 10 would only yield 20TB of storage (50% capacity utilization), while RAID 5 would give you 32TB of storage (80%

capacity utilization). And as more drives are added a RAID 5 array, this the capacity utilization improves. For this example, doubling the number of disks to 10 would give you 72TB in RAID 5 (90% capacity utilization). With RAID 5 you can use more of the capacity you pay for, while also enjoying performance gains and some level of redundancy.

But all this comes at a cost. The controller must work much harder to constantly distribute parity files across the array. Because of this, software controllers are usually not sufficient for good performance on RAID 5, which means you will have to pay more for dedicated hardware controllers. Also, RAID 5 is not for the lazy or technically naive. These arrays need careful attention, because there is only one point of fault tolerance. A failed drive must be replaced very quickly, or else the entire array is in jeopardy. Even if you do catch a drive failure and replace it immediately, the rebuild times of large arrays can be lengthy, so if another drive fails during that process, you will have completely lost *all* of your data. RAID 5 is definitely a cost-effective option that delivers performance and redundancy at scale, but you need to be understand the risks and technical obligations such a configuration entails. If you are prepared for that, then RAID 5 is a good option for media servers and databases in distributed production settings.

RAID 6 is a striped array with double distributed parity. It works almost identically to RAID 5, but it writes 2 parity copies of every file across the array rather than just 1. That means it can suffer 2 drive failures without total data loss, while also giving you better capacity utilization. RAID 6 is also more practical than RAID 5 for arrays with a large number of drives, because double parity removes some of the stress of long rebuild times. That said, RAID 6 still requires expensive hardware controllers to achieve consistent performance, and you will need to attention to its operation to spot drive failures. If you do that, then RAID 6 is an excellent option for media servers, databases, and other tasks that work need high-availability

data.

## Workflow Considerations

RAIDs are awesome. They give you more performance, more capacity, and more redundancy than single drives. Odds are, you'll use lots of different RAIDs in almost every stage of your workflow. But keep this in mind: no single RAID is a backup. Period. Why? Because a RAID is still one device. A RAID 1 with 10 disks might store 10 copies of your data, but it still keeps all those drives in a single enclosure. Yes, in this case your data is protected against drive failure, but it is not protected against someone knocking that enclosure off your desk. If that happens, you'll still lose everything, no matter how many copies there are inside the box. So, treat RAIDs like they're supposed to be treated; as a single drive. And for goodness sake, don't put anything on a volatile RAID (like RAID 0 and RAID 5) without keeping a backup somewhere else first.

## Hard Drives

Post-production environments require stable, large-capacity, high-performance storage. Playback response, file transfer times, transcoding, rendering, media access by multiple users (or not) are all affected by your choice of storage. But which choice is right for you? Let's start with hard drives.

When referring to hardware, *hard drives* are more broadly classified as *persistent storage*. Persistent storage is where data (e.g. files) is created, written, read, then stored for later use after your computer shuts down.

Sometimes, hard drives are confused with *memory*. As in, "I

need to back up my files! My computer's running out of memory!" What's the difference? With memory, or Random Access Memory (RAM), data is stored and referenced temporarily for rapid retrieval. Unless that data is written to a file in persistent storage, it won't stick around after the host computer shuts down. So memory and hard drives, not the same thing.

Back to hard drives. Here are two types:

1. HDD – Hard Disk Drives
2. SSD – Solid State Drives

An Overview of HDD's vs SSD's

Hard Disk Drives consist of disks attached to a motorized spindle (the "hard disks") working in concert with a magnetic head on an actuator arm. When the HDD is in use, the disks spin up (at thousands of RPM) while the actuator arm seeks a target position on the platter. Once reached, the magnetic head on the arm writes or reads data to the platter, repeating that motion until the operation is complete.

In Solid State Drives, mechanics are quite different. Instead of spinning platters and magnets, solid state chips are used for storage, while an advanced controller writes and reads data to those chips. In essence, the underlying tech used for RAM is now applied for persistent storage.

In general, SSDs are much more expensive, somewhat more durable, and a lot faster. HDDs are generally cheaper, but they can be slower and more prone to failure because of all of the fast-moving components.

Solid State Hard Drives are hybrids of HDD's and SSD's. A smaller capacity SSD and a larger HDD work with an

operating system to provide fast access to a user's most frequently used files. Eventually, the OS will have to use the much slower HDD for accessing files. Given the inconsistent performance of SSHD's and their reliance on the operating system to function as intended, they're not recommended for media drives, but make perfect boot drives.

## Key Attributes

The one thing you know: you'll need a drive with plenty of space (or capacity). If you're comparison shopping, you'll encounter even more information. What does it all mean? Does it *really* matter? Chances are, it will. Let's demystify some of these terms:

Capacity – the amount of data a drive can store. Available in quantities of:

- Gigabytes – 1 GB = 1000 Megabytes
- Terabytes – 1 TB = 1000 Gigabytes
- Petabytes – 1 PB = 1000 Terabytes

Sustained throughput – the amount of data that the hard drive can read or write continuously.

Interface – the physical connection of the drive to the host computer.

- This is important because the interface can be a bottleneck for the data moving on and off of the drive, affecting *sustained throughput*.

RPM's (HDD Only) – Revolutions Per Minute. This is how fast those platters spin in an HDD. More RPM's mean faster media access.

Drive Cache or Buffer (HDD Only) – like RAM in a computer, an HDD has its own bit of memory to quickly read and store data. The greater the cache size, the more data can be stored there, which means more media access performance gains.

Sequential Reads and Writes (SSD Only) – the possible *sustained throughput* of an SSD for either of these disk operations.

## Reliability & Drive Failure

A good indicator of drive reliability can be found easily: the manufacturer's warranty. If the warranty is 3 years or more, that's a good sign of their confidence in the product.

Unfortunately, drive failure isn't a matter of *if*, but *when*. Most drives come with S.M.A.R.T support, Self-Monitoring, Analysis and Reporting Technology. With SMART support, a drive may notify you of impending failure before it occurs. Many times, that doesn't happen, and SMART errors aren't the whole story.

## Sustained Throughput (Deeper Dive?)

Think of water as it travels through a pipe. The bigger the pipe, the more water can travel through it. That's the idea behind sustained throughput. The faster your hard drive is, the more data you can read or write per second.

Between large media files and demanding timelines, choosing storage that can handle a high volume of sustained throughput at any moment is another piece to maintaining momentum throughout your project and delivering on time.

It's also vital to understand your entire data pipeline to make sure that there are no bottlenecks. You might have a huge

pipe that can handle a ton of water, but if you screw a tiny hose onto the end, then the water won't come out very fast. So, you might have a super-fast SSD, but if you connect it to your computer via USB-2 (the old USB standard), then your data is going to move very slowly.

Here's a quick key on bitrates:

- Mb/s = 1 Megabit per second
- MB/s = 1 Megabyte per second = 8 Mb/s = 8 (bits makes a byte) * 1 Mb/s
- Gb/s = 1 Gigabit per second = 1000 Mb/s

## Interfaces

A hard drive's interface is the physical connection to the host computer, and an indicator of possible sustained throughput during its operation.

Here are some common interfaces and their maximum supported throughput:

Serial ATA or SATA (Internal or External)

1. III – Up to 6 Gb/s
2. II – Up to 3 Gb/s
3. I – Up to 1.5 Gb/s

Thunderbolt

1. 3 – Up to 40 Gb/s, bi-directionally
2. 2 – Up to 10 Gb/s, bi-directionally

3. 1 – Up to 10 Gb/s, bi-directionally

Ethernet (often labeled in Gbps or Mbps)

1. 10 Gbps
2. 1 Gbps
3. 100 Mbps

USB

1. 3.1 Gen. 2 – Up to 10 Gb/s
2. 3.1 Gen. 1 – Up to 5 Gb/s
3. 3.0 – Up to 5 Gb/s
4. 2 – Up to 480 Mb/s
5. 1.1 – Up to 12 Mb/s

Firewire

1. 800 – 800 Mb/s, bi-directionally
2. 400 – 400 Mb/s, bi-directionally

## RPM's (HDD Only)

Another factor to HDD performance is the RPM's, or Revolutions Per Minute, of the platter. The higher the RPM count, the faster your drive can access data. How fast do you need, though?

Anything less than 7200 RPMs is probably unsuitable for high-performance media drive use. If you're looking for a low-cost backup, archive, or general storage solution, then a slower

hard drive may be fine, but (generally) the higher the RPMs, the faster the drive will be.

It's not just the RPM's, though.

## Drive Cache or Buffer (HDD Only)

A drive cache, or drive buffer, is a bit of memory bundled with a HDD. If the HDD has a substantial cache size, more data can be written, read, and accessed from this high-speed storage point instead of its platters. Remember, an HDD is a series of moving parts working together as fast they can. But the choreography of platters, arms, and magnetic heads won't outpace the sheer performance of solid state chips.

But what if the whole drive was a solid state chip?

## Sequential Reads and Writes (SSD Only)

Since SSD's contain solid state chips with no moving parts, forget what you know about RPM's and cache size. What you *will* see are listings for these details:

- Sequential Reads – Compressible & Incompressible Data
- Sequential Writes – Compressible & Incompressible Data

Those are followed by a number of data units calculated per second (e.g. 500 MB/s, or Megabytes per second). This is an indicator of the possible sustained throughput during its operation. Remember: 1 Megabyte per second = 1 Megabit per second multiplied by 8.

Let's put this all together.

## Hard Drives and Video Playback

Let's say you're working in Avid Media Composer, and you're working with footage transcoded to the DNxHD 145 codec. With DNxHD, the number is the bitrate of the media, or the size of the water moving through the pipe, measured in megabits (not bytes) per second. So a file encoded in "DNxHD 145" means 145 megabits per second of throughput will need to travel between the drive and its host computer.

Once that clip's inserted in your timeline, it becomes one stream of video occupying around 145 megabits on playback. Insert another clip on top of your existing clip, and one stream becomes two at around 290 megabits per second of throughput How many streams can a hard drive handle?

Let's go back to that SSD rated at 500 megabytes (not bits) per second of throughput. Since a DNxHD 145 file's bitrate is 145 megabits per second, we'll convert the SSD's rated throughput of megabytes per second (MB/s) to megabits per second (Mb/s):

Megabytes per second * 8 = 8 Megabits per second

500 MB/s * 8 = 4,000 Mb/s

So if one stream of DNxHD is 145 megabits per second, and the SSD is rated at 500 megabytes per second, we can convert the codec's bitrate to determine how many streams of video could be played simultaneously on a timeline from this drive.

(4,000 Megabits per second) / (145 Megabits per second) = 27 streams of video

On the other hand, if your files are 8K Redcode Raw at 1,216 Mb/s, that super-fast SSD will only be able to handle 3

streams, max.

Of course, all of this is assuming the drive is connected to a host computer with an interface that allows the maximum amount of throughput. But what if someone decided to install that SSD in a USB 2 enclosure? Or what if it's a part of shared storage appliance connected over Ethernet at 1 gigabit per second? All of these are potential bottlenecks to your desired sustained throughput.

This is why many editors love to cut using low-bitrate proxies like DNxHD 145–because the data speeds are very manageable without high-performance hard drives.

## Choosing the Right Hard Drives

Again, postproduction requires stable, high-performance storage. Usually it comes down to budget, but SSD's and premium HDD's have become more and more affordable. Weighing out all factors and deciding on the right storage will save you time in the edit, and help you deliver on time.

Contributed by Isaac Terronez, Producer, video editor, and technologist.

## 1Gig Ethernet vs 10Gig Ethernet

If your video team is going to start working from a server or any sort of NAS, one of the most important things to decide is whether you're going to connect to your server via 1 Gigabit Ethernet (1GbE) or 10 Gigabit Ethernet (10GbE). What this really means to an editor is, based on the type of work you plan to do, can you get away with having a *pretty fast* connection of 100MB/s, or do you need a *really fast*

connection up to 1,000MB/s?

As a general rule of thumb, if you're workflow tops out at HD you can get by with a 1GbE connection. This even holds true for higher quality HD ProRes codecs like 4444XQ and multicam projects. Nearly every laptop and desktop out there can take a 1GbE connection by either plugging directly into it, or in the case of Apple laptops with a $15 dongle. Additionally, if your facility already has Ethernet cabling installed it's very likely 1GbE compatible cable (CAT 5 or 6) running through the walls. A 1GbE is also the perfect way to connect up graphic designers, VFX, and your audio team as their bandwidth requirements tend to be minimal.

Where 10GbE starts to make the most sense for your team is when you're consistently working with 4K footage. For reference 1GbE tops out at a single stream of 4K ProRes 422HQ in 24 frames per second while a 10GbE connection can handle 5 concurrent streams of 4K ProRes 4444 HQ at the same frame rate. Until recently connecting a Mac to a server over 10GbE required a special Thunderbolt peripheral that could cost upwards of $500, however Apple has started including a 10GbE port on their iMac Pro and even the latest Mac Mini which is proof that 10GbE is definitely the way things are headed. Worth noting is that in order to carry a 10GbE signal your Ethernet cable must be rated CAT6 or higher depending on your distance from the server.

At the end of the day it comes down to your need for speed, not only in the timeline but with ingesting and copying to and from your server. We can say that 10GbE is becoming more of a standard and if you're making a sizeable investment in your gear it's worth trying to futureproof as much as possible.

Here's a handy guide from Apple on ProRes data rates to help calculate your bandwidth requirements:
https://www.apple.com/final-cut-pro/docs/Apple_ProRes_White_Paper.pdf

Don't mix up your bits and bytes!

1Gb = 1000Mb = 125MB

10Gb = 10000Mb = 1025MB

Gb – Gigabit

Mb = Megabit

MB = Megabyte

Contributed by LumaForge.

Bin locking and simultaneous access

*Bin Locking* is the basic concept that allows for collaborative editing, and it is very simple in theory, though the details can become a little bit complicated. In essence, bin locking gives a way for one editor to tell the other editors "Hey, I'm working on this section. Don't change it until I'm done… ok, I'm done."

When multiple editors or assistants have access to the same set of files without a locking mechanism, it can cause a lot of problems. If two people are working on the same sequence at the same time, they may overwrite each other's changes without realizing it, causing confusion and lost work.

Depending on your workflow, it may be possible to do some basic "manual" locking of files. For instance, if you're using Premiere Pro and have a project file called cool_film_scene_5.prproj and you want to start editing it, you could change its name to

cool_film_scene_5_LOCKED_BY_SARAH.prproj. If anyone else wanted to work on that sequence, they would be able to tell by its file name that you're working on it. Then, when you're done, you change its name back to the original, and then someone else can lock it again with their own name and begin working.

This kind of workflow can be tedious, however, and can cause other issues if you're not careful, so it's usually better to use an NLE's dedicated tools for bin locking, if available.

## Bin Locking in Avid

Avid Media Composer was the first NLE to add a set of features for bin locking. Although other NLEs have started to implement some of those features, Avid's bin locking is still the gold standard and the most simple and reliable, so we'll go into some extra detail of Media Composer's bin locking tools.

With Bin Locking, your team has simultaneous access to a Media Composer Project, with rules in place to define access to Bins. Bin Locking is available when using Media Composer with one of Avid's shared storage products, or using third-party shared storage with Avid file system (AvidFS) emulation.

## Core Mechanics

### Defining Access Rights

Bins are opened in a Shared Project in one of two access modes, Write and Read-Only. Write access lets you work within a Bin as expected in a single-user Media Composer

environment – read, write, modify, and delete Sequences or Clips. Read-Only lets you open a Bin and its contents for viewing, but you can't modify or delete its contents.

### Defining Simultaneous Access

Simultaneous access is performed by checking out or checking in Bins. Enter Bin Locking.

Checking out a Bin with Write access *locks the Bin for use by you*. No one else can modify or delete this Bin's contents until you check it back in, *releasing the lock*.

If someone else attempts to check out a locked Bin, it opens with Read-Only access. If they try to modify or delete its contents, Media Composer will prompt them to save that Bin as another.

Media Composer also displays the current state of a locked Bin, and logs who may be responsible for the last set of changes saved to that Bin.

## Collaborative Editing Possibilities

Here are some collaborative possibilities addressed by Bin Locking.

When a Lead Editor is working with Assistant Editors, the Lead may check out (again, lock) a Bin because he doesn't want the Assistants modifying it. But what if the Assistants need *that* Bin for their own tasks? Although opened and locked by the Lead, the Assistants can open that Bin with Read-Only access. Then they can to create their own Bin opened with Write access, copying any Sequences or Clips needed from the locked Bin

to their new one.

Conversely, if an Assistant's gathering and logging assets for a Lead with the intent of copying those assets to a locked Bin, the Lead will have to check the Bin back into the project, releasing the lock. If the Assistant has Write access rights, they can open that Bin and continue.

What if a Producer wants to see a version of a Sequence without breaking an Editor's train of thought? If the Editor is working on that sequence in a locked Bin, a user with rights to the Project can open that Bin in Read-Only mode, then play back the Sequence in that Bin for the Producer.

## Troubleshooting

A common caveat: sometimes the lock, as represented by a .lck file in an Avid Projects folder, isn't released or is corrupted. Troubleshooting may be a coordinated effort to locate the corrupt .lck file, delete it, then verifying if the Bin is available again.

With its familiarity and defined access structure, Bin Locking is a time-honored approach to enabling collaborative editing workflows in multi-user, Avid Media Composer-centric environments.

Contributed by Isaac Terronez, Producer, video editor, and technologist.

# Offline Editing

We mentioned briefly above that the files that come directly out of the camera aren't well suited to editing, so it's very common to convert those files into a more edit-friendly codec before the editing process starts. This process of working off of a separate set of files is called an offline edit, and the files generated specifically for the offline edit are called proxy files (or proxies).

How to choose a proxy codec

When choosing a proxy codec, you want to go for one that does not use temporal compression (aka inter-frame compression or long-GOP compression), and you want to pick one that has a lower bitrate. The low bitrate means that the files are much smaller, so you can use fewer/smaller/cheaper hard drives, simplifying your workflow.

By far the most common proxy codecs are DNxHD/DNxHR and ProRes. They have both been around for years, so they're very widely supported. Everyone knows how to handle them. They are both very well suited to a proxy workflow (ProRes even has a preset called "proxy"), and are nearly interchangeable when used for proxies.

Since DNxHD is made by Avid, and ProRes is made by Apple, it makes sense that DNxHD would work better on Media Composer and ProRes would work better on Final Cut Pro X. That used to certainly be true, but nowadays both codecs work very smoothly on all modern editors (including Premiere Pro and Resolve). There may be a slight speed increase in using the codec that was designed for the system, but it's very slight.

The only significant difference between the two for a proxy workflow is the fact that you may have trouble creating ProRes

on a PC, while DNxHD is very easy to create cross-platform. PCs can easily play back and edit ProRes files, but you can't encode new ProRes files on a PC as easily as DNxHD, and so some editors prefer a DNxHD workflow for that reason.

A third good option is Cineform, which was acquired by GoPro but is still supported by all of the creative tools. Cineform offers similar options to DNxHD and ProRes, and it has shown slightly better image quality in some tests, but not by a huge amount.

Regardless of which of the codecs you pick, you also have to pick which flavor you want. This is really going to depend on your storage constraints – it's a tradeoff between image quality and file size. The good news is that you don't need tip-top image quality when you're editing, so you can choose a low-bitrate codec.

Start off with the smallest ProRes or DNx codec in the same resolution as your capture codec. Look at the GB/hr column and multiply it by the number of hours of footage you have. If you have enough storage space, then you're good – use that codec. If you have lots of extra storage space, think about using the next largest flavor.

If you don't have enough storage space, or if you're on an underpowered machine, then take the resolution down a notch. A lot of huge-budget Hollywood films were edited in 480p just a few years ago, so don't sweat it if you need to lower your resolution from 4K down to 720p for the edit.

Of course, you always want to edit with decent-looking footage, but it can be hard to estimate which codecs and bitrates provide the clearest image. Check out the Codec Comparison to see how different codecs impact image quality.

While the proxy files are great for editing, you shouldn't do more than basic color-correction with proxy files. If you are going to do all of your color-correction inside of your editing software, then it's best to re-link back to your camera files because your proxy files more than likely have lower color quality.

The good news is that most editing software today can switch between the camera files and the proxy files in just a couple clicks, so you can even go back and forth if you need to.

We've written detailed guides for proxy workflows in each of the major NLEs:

Final Cut Pro X Proxy Workflows

Proxies give you tons of flexibility – they increase playback and editing speed without forcing you to sacrifice quality, allow you to edit seamlessly on laptops and portable drives, and provide a streamlined project delivery, whether that includes a hand off to a mixer or colorist, or exporting a full quality output for a client. And FCP X makes all of that simple.

Say what you will about its initial release back in 2011, there's no denying that FCP X was a leader in redefining the proxy workflow – it was the first of the big NLEs to allow for creating and switching to proxies with the click of a single button, all within the application.

At its core, FCP X's proxy workflow remains incredibly simple – creating proxies is as easy as checking a box upon import.

But because of FCP X's unique library structure, how to take proxy media on the go remains a big source of confusion among editors. With a little media management know-how, anybody can take advantage of proxy media's portability.

By the end of this article, you'll have a complete understanding of how to manage your FCP X library for proxy success, how to create proxies during or after import, and how to take your portable proxy media on the road.

## What are Proxies?

Proxies are lower-resolution copies of your media that are created in one of two ways – either by transcoding raw media to an intermediate codec or by simultaneous in-camera recording, a standard feature on many high-end cameras.

Editing with proxy media is commonly referred to as an "offline edit," while the process of relinking back to the full-res media at the end of a project is called the online edit. FCP X's proxy workflow essentially combines the two, allowing you to effortlessly switch between offline and online with one click.

A previous article explains in detail why using proxies is awesome and how they fit into an overall postproduction workflow. Here, we're going to dive into the details of how FCP X deals with proxies.

So how do we get started? First, we need to understand how FCP X manages our media.

## FCP X Library and Media Management Overview

If you're an FCP X editor, you likely have an understanding of how FCP X references and manages media. But a successful proxy workflow requires a specific storage setup, so let's review.

In FCP X, you organize your work in libraries. A library contains events and projects and tracks all your media files, edit decisions, and the associated metadata.

Let's take a closer look:

- Libraries are the top-most level of organization and hold all of your events, projects, and often, your media. You can have as many libraries as you'd like, with the option to open and close them within FCP X as needed. You can save a library wherever you'd like and can migrate events and projects from one library to another. Libraries contain all of the information relevant to your edit, wrapped into one file on your hard drive. They're similar to what other NLE's call project files, except that libraries are capable of holding your media as well.
- Events are stored within libraries, and every library must contain at least one event. We find that it helps to think of events like folders, since events hold and organize your media within your library. For example, on a feature film, you may create separate events for each shoot day's dailies. For a short form project, you may label events based on camera media cards. You may also have separate events for your music or graphics.
- Projects are the lowest level of organization in FCP X and are housed inside events. "Project" is synonymous with the terms "timeline" or "sequence" used in most other NLE's.

By design, FCP X can manage all of your media for you, including camera media, proxy media, render files, and anything else you import, in a single place – the library.

This is a powerful feature and can be useful in many situations, like if you intend to edit your project on a single computer or hard drive from start to finish. And if that is your situation, go ahead and adapt your storage setup to meet your own needs.

But we want the freedom to both take our smaller, lightweight

proxy media on the go to and to work off a lower powered laptop without sacrificing performance. To do that, we need to retain control over the locations of our original camera media and our proxy media.

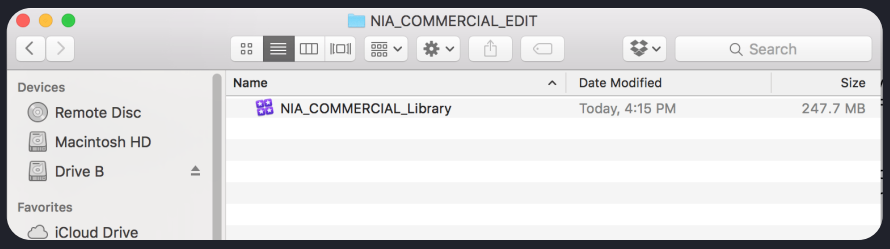
FCP X recognizes these different storage needs and allows for two types of media management:

- Managed media refers to media that you let FCP X manage for you. In most cases, this means that you copy your media into FCP X's library structure and let it handle the organization from there.
- External media refers to media that you manage yourself. In most cases, this means that you organize your media on your hard drive as you see fit. When you import it, you tell FCP X to simply link to it, rather than copy it into the library structure. If you're familiar with Adobe Premiere or have used FCP 7 in the past, then you're likely more familiar with external media management.

To fully understand this concept, let's break down the structure of an FCP X library.

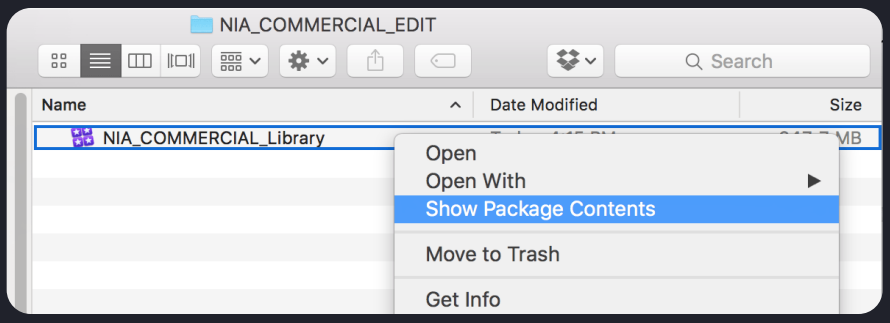
## Library File Contents

This is what an FCP X library file looks like on your hard drive.



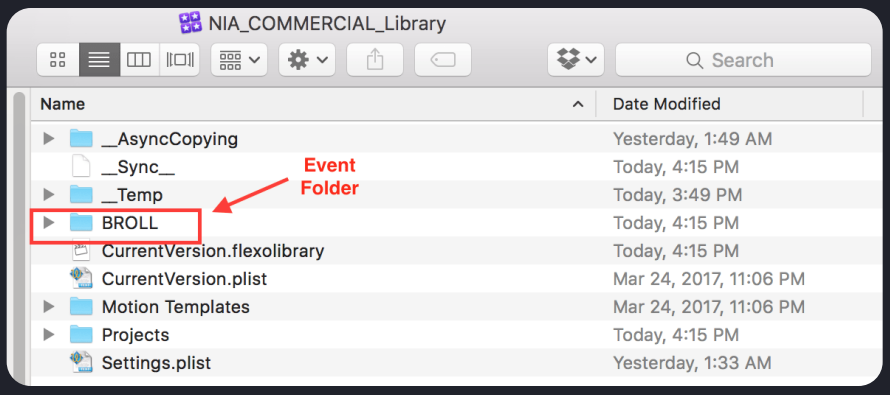

Though it looks like one single file, it actually contains many other folders and files within it. If you right click on a library file

and select Show Package Contents from the context menu, the library will open like any other folder.

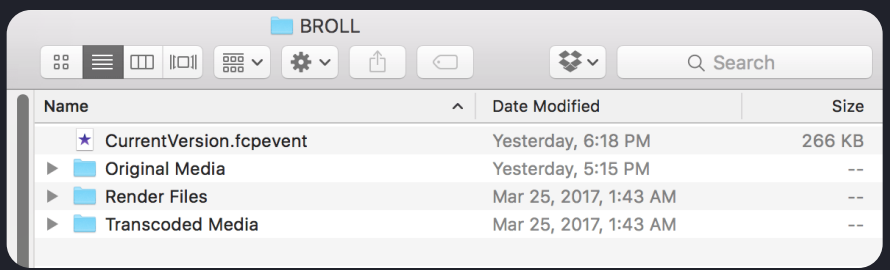



Inside, we can see a few metadata files and folders that we generally don't need to touch.

We're interested in the corresponding folders for each event that we created in FCP X. In this case, there is one event called BROLL that was created in an FCP X edit and then imported all the broll footage into it. If you have an event called Wedding Ceremony in your edit, then you'll find that folder here as well.

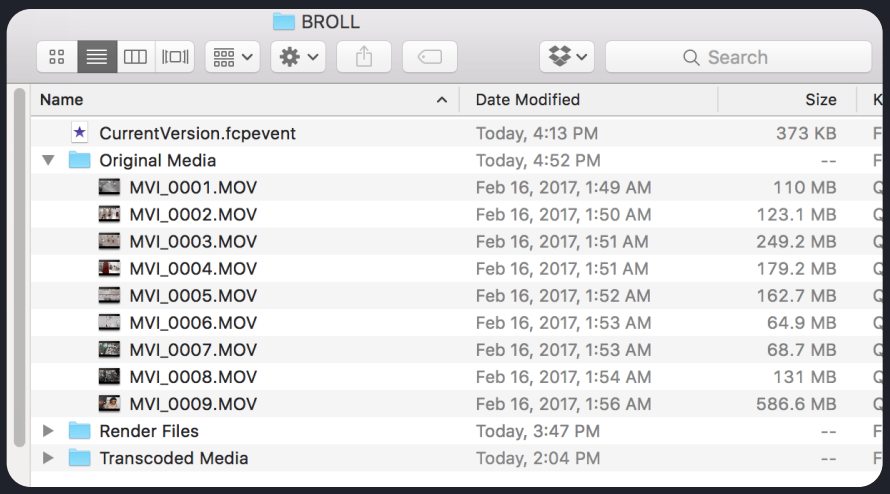



If we open up any event folder, we'll see additional folders for Original Media, Render Files, and Transcoded Media.

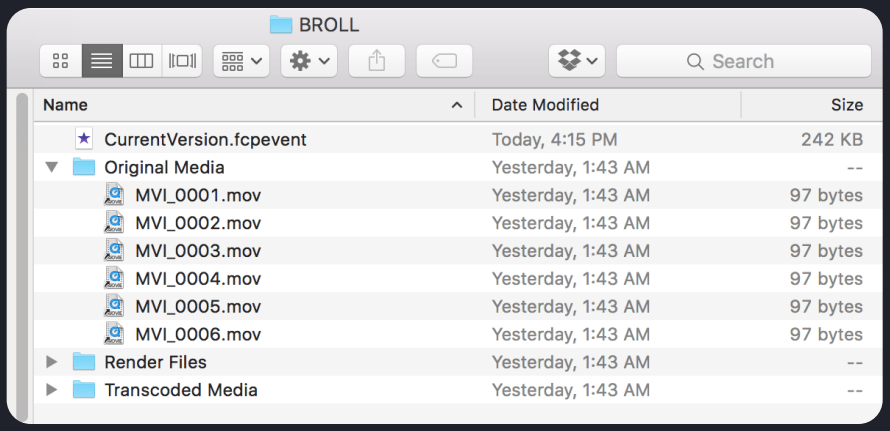


The Original Media folder is where FCP X stores our imported media. The Transcoded Media folder is where FCP X will eventually store our proxy files.

This is an image of a managed library. All of the camera media is right here, in the Original Media folder.



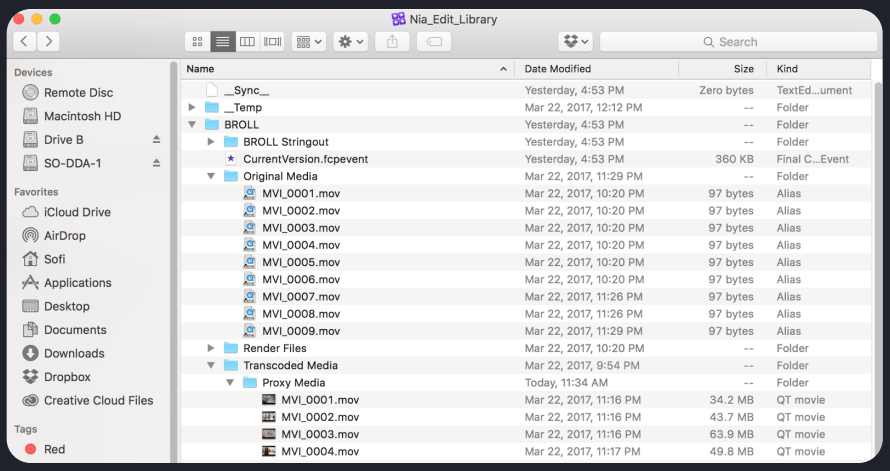

By contrast, this is an image of a library with external media.




Note the little arrow icons in the bottom left corner of our media clips. Those icons mean we're looking at files called symbolic links or symlinks for short. Symlinks tell FCP X that our media is stored externally and act as pointers to that location. You can also tell that these are pointers rather than the actual media, because they are very small files (97 bytes).

To give us ultimately flexibility for our proxy workflow, we're going to use a combination of external media (for our high resolution camera files) and managed media (for our proxy files).

Ultimately, we want our library to look like this, with external media in the Original Files folder and managed media in the Transcoded Media folder.




For now, let's focus on our original, high resolution camera media – our external media.

For most professional level projects, maintaining the camera media's original file structure is imperative. Breaking that file structure can be detrimental for sharing edits between FCP X and other programs like DaVinci Resolve.

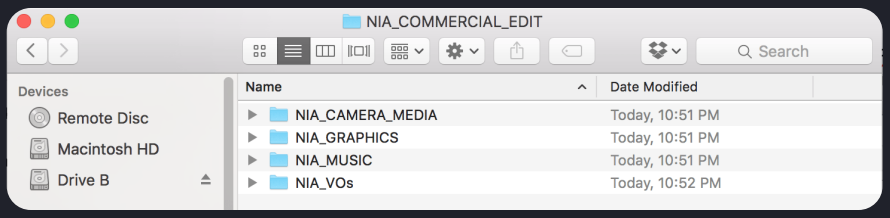
We don't want FCP X to manage this media for us unless we're sure that the entire project will only ever live in FCP X.

## Organize Your Media on Your Hard Drive

Before we even open up FCP X, let's do a little media housekeeping. First, we strongly suggest you organize your

entire edit into a single container folder, and then organize all of your camera media into its own container folder.

For example, the footage you'll see in the screenshots throughout this article comes from a commercial shoot for a fashion line called NIA. The top-level folder is named NIA_COMMERCIAL_EDIT and immediately inside, the camera media container folder is named NIA_CAMERA_MEDIA. All of the camera media gets copied directly into this folder, with as many subfolders for shoot dates, camera cards, pickups, etc., as needed.



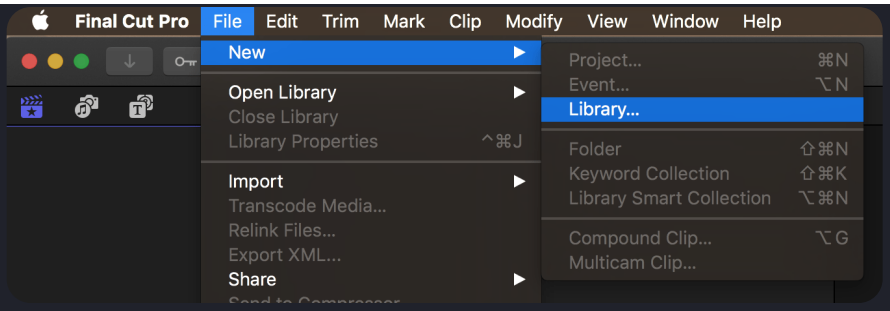

Generally, you should have similar folders for graphics, music, voiceovers. That way you know where everything is at a glance.

Once you've got your camera media organized, it's time to create the library that we'll work out of.

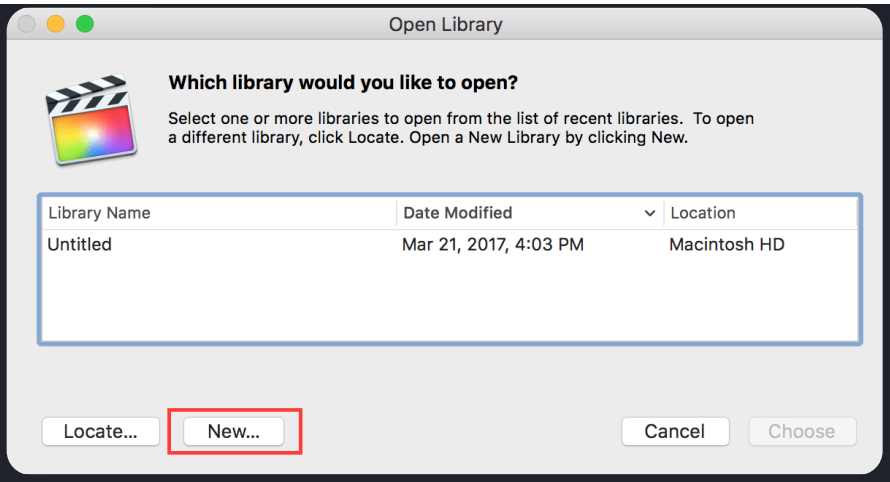
## Creating a Library and Setting up Storage Locations

Launch FCP X and close any currently open libraries to avoid confusion.

To create a new library, navigate to File > New > Library.



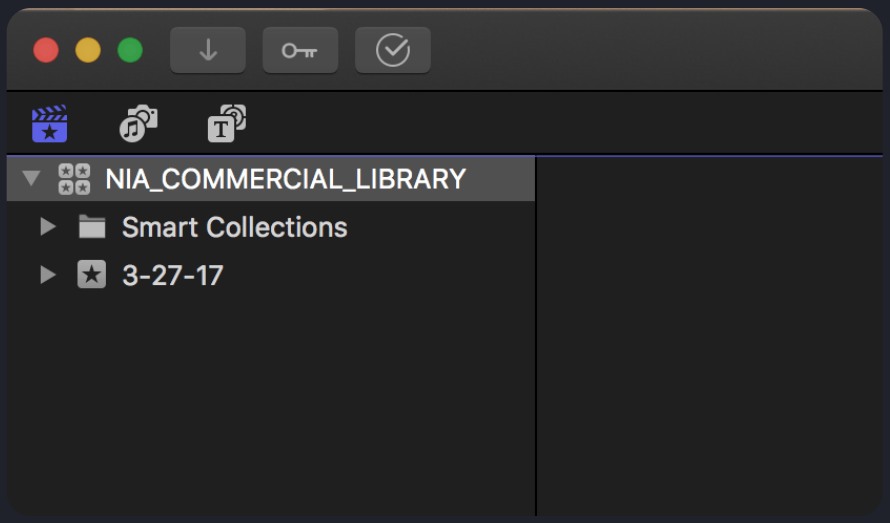

You can also click the Open Library button in the sidebar and select New in the Open Library dialogue box.




In the Save dialogue box, name your library and then navigate to your project's container folder. You don't have to create a new folder to house the library file because, as we've seen, it acts as its own folder, keeping all of FCP X's data inside, but you can if you'd like to.

When you're ready, hit Save.

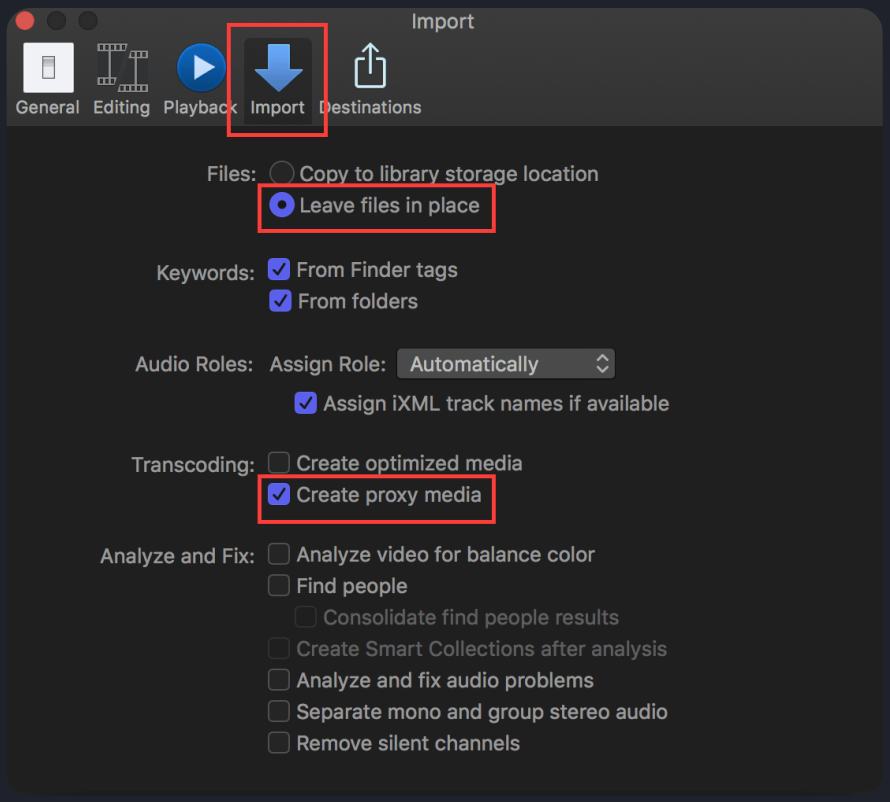
In the sidebar, you'll now see your new library, pre-populated with a Smart Collections folder and an event labeled as today's date.




Now that we have a library, we can start importing footage.

But first, it's important to set our import preferences. Use the

keyboard shortcut "⌘," or Navigate to Final Cut Pro > Preferences to open the Preferences window.




In the Preferences window, click on the Import tab. Under the Files heading, we have two options:

- Copy to library creates managed media. By default the media we import will be copied from its original location to be stored inside the selected FCP X library. I'd like to note that we can actually tell FCP X to copy our media to a location outside the library, but when we do this, FCP X reorganizes our media into its own folder structure, rather than maintaining the original file structure, which is not what we want.
- Leave files in place creates external media, meaning that the imported media remains in its original location and is linked to the library with symlinks, rather than copied over to it.

We want to choose Leave files in place. The other Import setting we're concerned with is under the Transcoding heading, where we have another two options: Create

optimized media and Create proxy media.

We're mostly concerned with the second, but there are situations where optimized media is useful, so let's go over both.

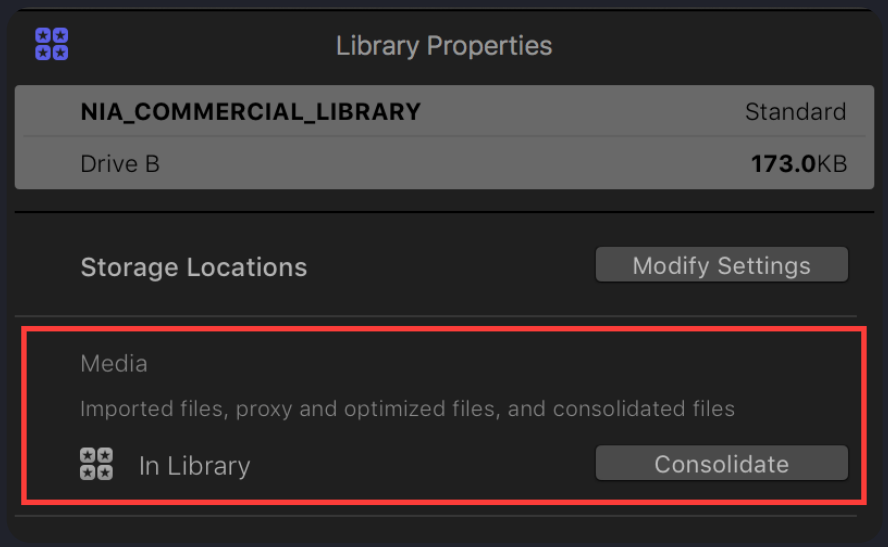
- Create optimized media: On import, transcode your original media to the ProRes 422 codec, allowing for better editing performance and faster render times. Because this option will maintain the same frame size as your original media, we don't think of this as proxy media, but more as a direct intermediate. This option is most useful if you're ingesting footage with a frame size that your computer can handle well, such as 2K or smaller, but in a codec that's not edit-friendly, such a highly compressed .mp4. By creating optimized media, you can significantly improve editing performance and speed, while maintaining your original frame size. In many cases, these optimized media files will be high quality enough to serve as new master clips. Note that if your original camera media codec can be edited with good performance in FCP X, the create optimized media option is dimmed.
- Create proxy media: On import, transcode your camera media to the lightweight ProRes 422 (Proxy) codec at one-half the original frame size, creating smaller files sizes for faster editing performance and portability. With this option checked, still images you import are transcoded to either JPEG (if the original file doesn't have alpha channel information) or PNG files (if the file has alpha channel information). Note that the codec and frame size for proxy media can't be customized.

Check the Create proxy media box and close the Preferences window.

Based on these selections, we know that when we import our camera media, it'll stay in its original location and be linked to our library. We also know that we'll be creating proxy media on import, but where does that go?

To find out, let's take a look at our Library Properties. In the sidebar, select your library. Now open the inspector by using the keyboard shortcut "⌘4" or navigating to Window > Show in Workspace > Inspector.

At the top, the inspector lists the name of our library and underneath, what drive it's stored on and how much space it's occupying. Below that, we can see a summary of where our media and other files associated with our library are stored.




If we take a closer look at the Media section, we can see it lists that imported files, proxy and optimized files, and consolidated files are set to be stored In Library.

Don't let this information confuse you – once we set our Import Preference to "leave files in place," our camera media won't be copied to the library, even though it doesn't specify that here (I wish it did!). The important information is that our proxy files will be stored in the library.

Remember when we said we'd be using a combination of managed and external media? We're going to keep our proxy files as managed media – meaning they'll be stored inside the library.

Why? In a nutshell, FCP X has two serious limitations that when combined, form a perfect storm that can destroy its own proxy workflow.

- First, while it has tons of options for moving and copying media, events, and projects, FCP X has no option to move or copy ONLY proxy media. The only way to move or copy proxy media from within FCP X is to move or copy the original camera media along with it – something we'd almost never want to do.
- Second, FCP X has no option to relink offline proxy media. That means that if we try to circumvent the above issue by managing proxy media externally (outside of the library), and we copy that proxy media to a portable hard drive, FCP X will not be able to relink to the proxy media in its new location.

To keep things as simple as possible, while leaving as little room for error as we can, we keep our camera media as external and our proxy media as internal.

That way, we prevent proxy media from going irreparably offline by letting FCP X manage it inside the library, and we bypass copying over original camera media by managing it ourselves, outside the library.

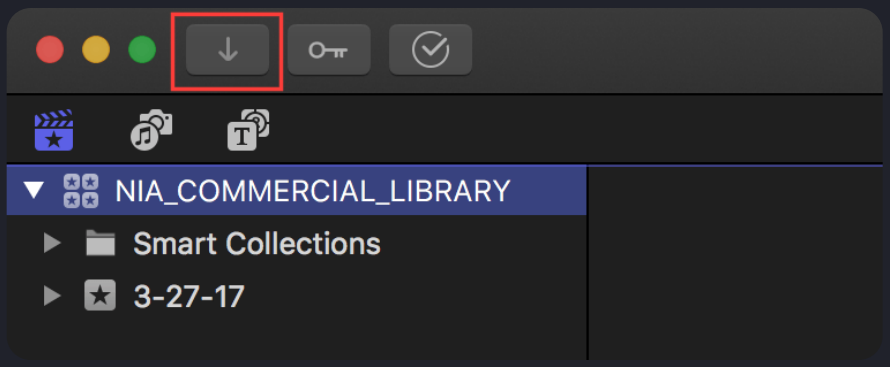
To bring this all together, let's import some media, create some proxies, and take a closer look.

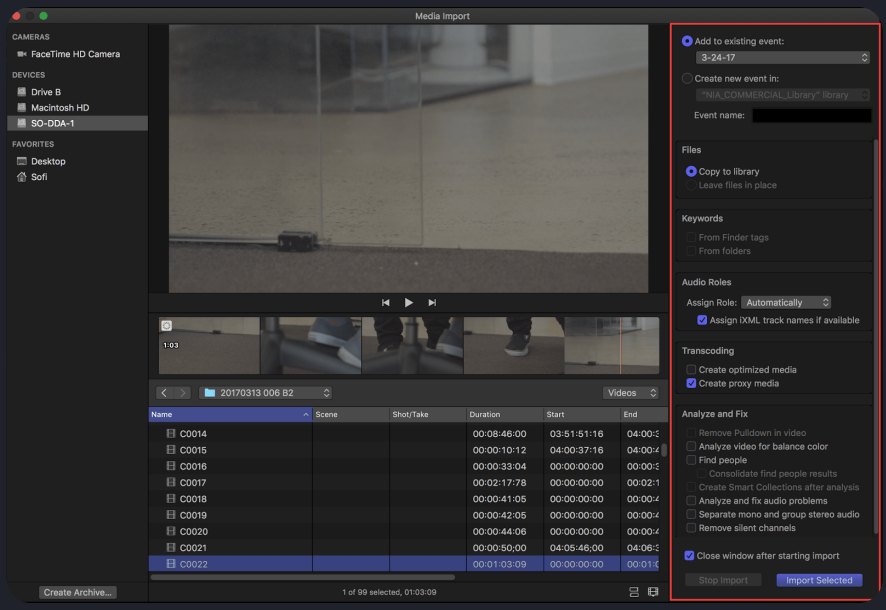
## Importing and Creating Proxies

In FCP X there are two ways to import media: through the Media Import window or by dragging and dropping into an event.

# Importing Through the Media Import Window

Selecting an event and clicking the Import Media button or using the keyboard shortcut "⌘I" will bring up the Media Import window.



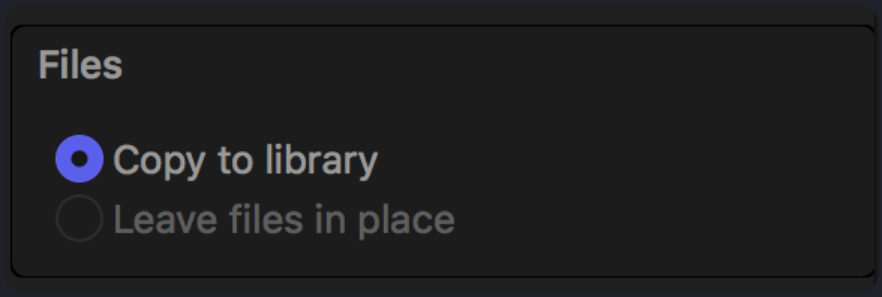




In addition to allowing you to navigate to your original media, you'll notice that on the right-hand side, we have the same settings available in Import Preferences. These settings are all global, meaning that we can set them here or in Import Preferences and they'll remain set in both places. Except for one.

The Media Import window has the ability to override the "Leave files in place" setting and for that reason, we don't

recommend using it.

When browsing files through the Media Import window, FCP X automatically detects file structures that resemble camera media, and when it finds this type of media, it greys out the "Leave files in place" option forces you to choose the "Copy to library."




Why? FCP X thinks that you're copying media from an actual camera memory card, so it forces you to copy that media over to the library to stop you from accidentally editing off of a memory card.

It's true that you should never edit from removable media that will likely be ejected and erased. However, as we've mentioned several times, in a professional workflow it's always best practice to leave the original camera media's file structure intact, meaning that if you use the Media Import window, FCP X will always think you're importing from a real camera card, regardless of whether you are or aren't.

In future releases, let's hope this feature changes to an overridable warning, rather than an unalterable default.

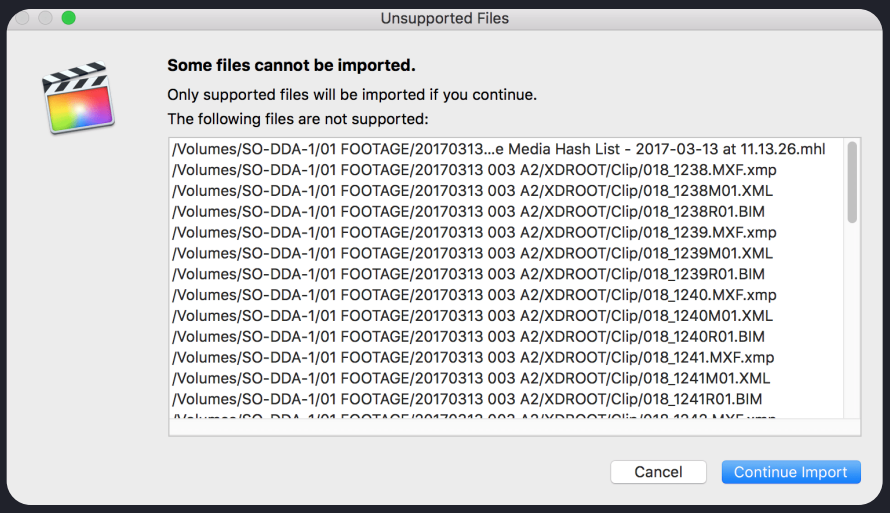
## Drag and Drop Importing

Drap and drop importing bypasses the Media Import window's memory card issue, allowing you to import your

media and maintain all of the settings you've selected in Import Preferences.

For the most part, you can drag the highest level folder containing your clips right into your event and all of the media inside will import.

Be aware that you'll usually get an Unsupported Files message, alerting you that some of the files within those folders won't import, such as the camera's metadata files. That's fine, though. We don't need those files in FCP X anyway.



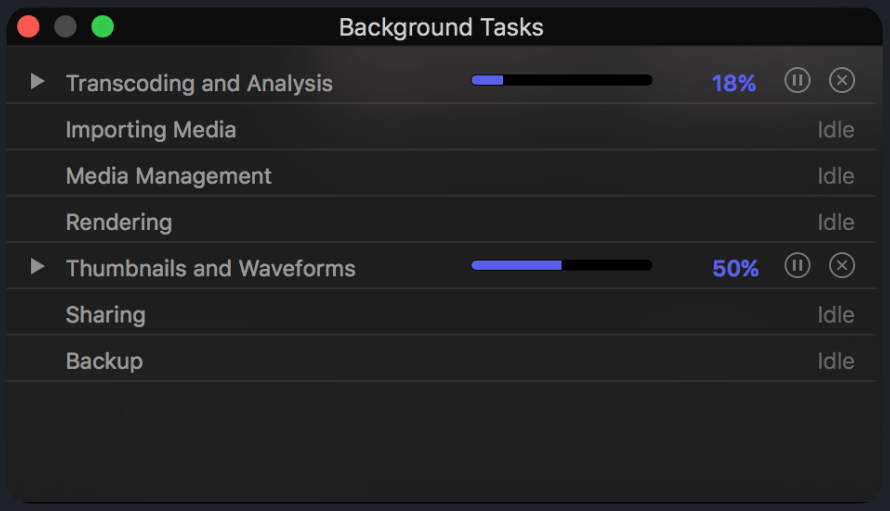

As soon as you import media by dragging and dropping, FCP X gets to work creating your proxy files.

## Background Processing and Monitoring Progress

FCP X creates proxies using background processing, meaning that you can keep working, labeling or even editing your clips, while the proxies generate in the background. As they finish, they'll be automatically attached to your high res clips.

You can monitor the progress of all background tasks by

using the keyboard shortcut "⌘9", or navigating to Window > Background Tasks.



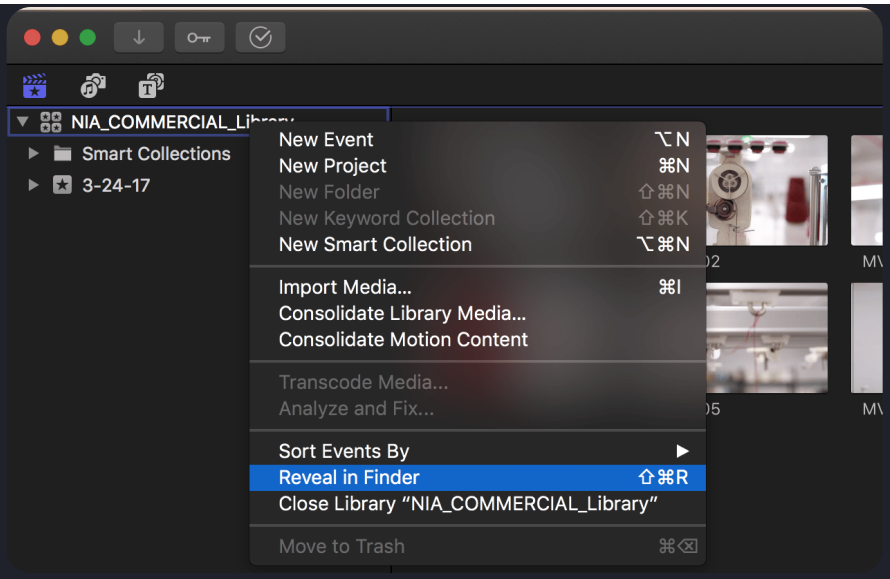

The Background Tasks window lists a variety of processes that may be happening behind the scenes. The first section, Transcoding and Analysis, is where you can see the progress of your proxy clips.

## What if You've Already Imported Media?

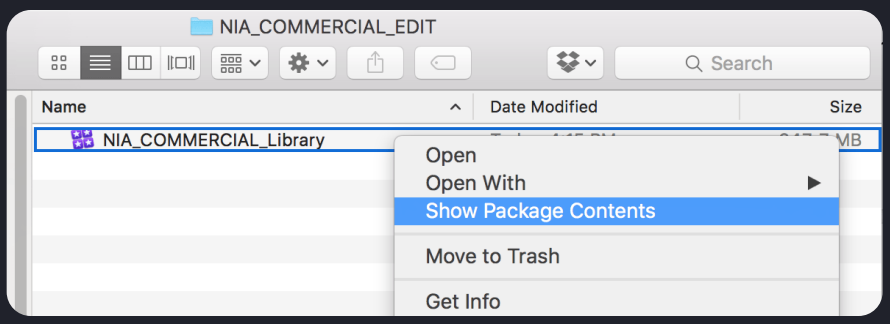
If you've already begun editing or importing without having created proxies, don't worry. You can also create them after import, right from the browser.

Before you do anything else, STOP and consider whether your original imported media is external or managed. If you want to take your proxies on the go, your imported media must be external.

Not sure? Select your library in the sidebar, right click it and select Reveal in Finder from the context menu.




This will take you to where your library file is stored on your hard drive. Right click the library file and select Show Package Contents.




Navigate to and open one of your event folders, and then open the Original Media folder. Are the media files inside symlinks? If so, then your media is external. If you have multiple events, it's a good idea to run through them here and make sure all of their media is external.

If you don't see symlinks, you have managed media and need to make it external before creating any proxies.

We can do this from within FCP X using the consolidate feature.

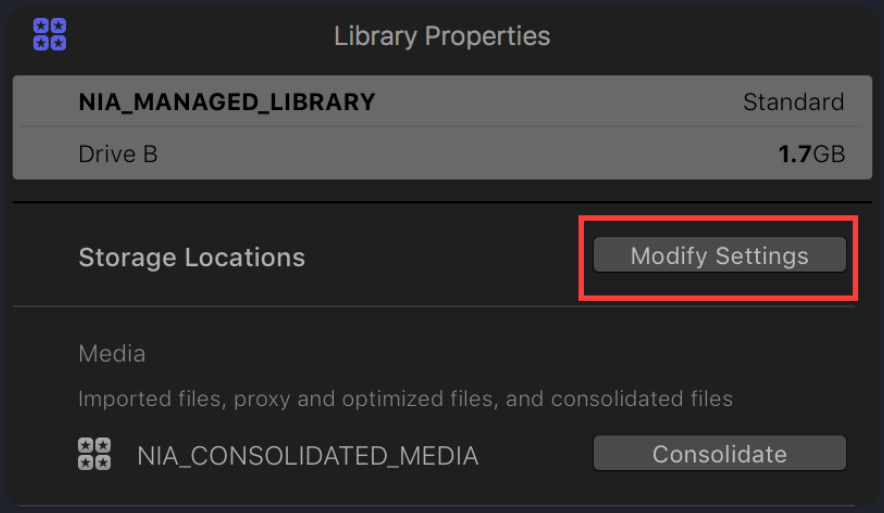
# Consolidating to Create External Media

Consolidate allows us to gather all the media in the project and move it to a new location. We can use it to move our managed media to a folder outside the library, making it external.

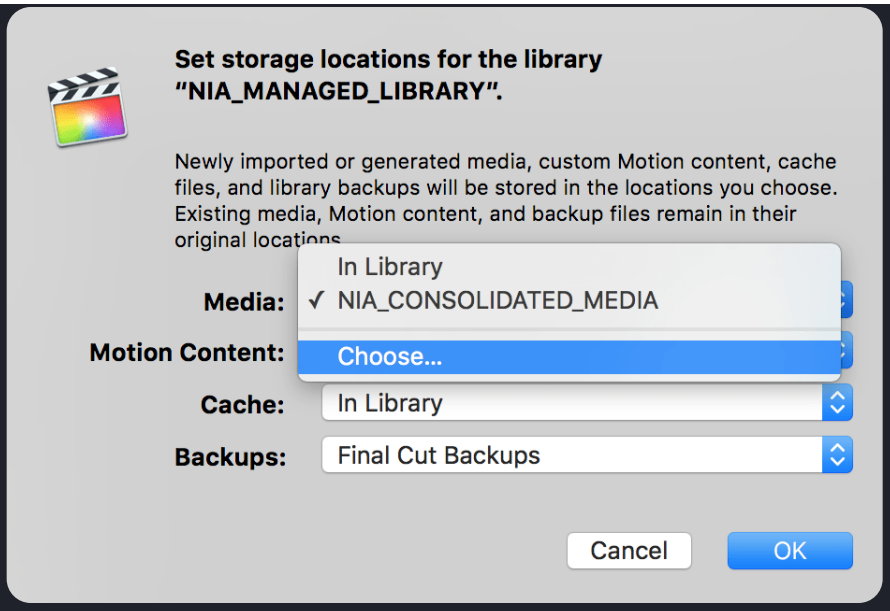
It's important to note that ALL media associated with your library will be copied to the new folder with this method. For example, if you've imported your camera media as managed, but have kept your music as external, the consolidate feature will copy both the camera media and music to the new location.

To consolidate, select your library in the sidebar and if it's not already visible, open the inspector panel using the keyboard shortcut "⌘4" to bring up Library Properties.

Select the Modify Settings button next to Storage Locations.

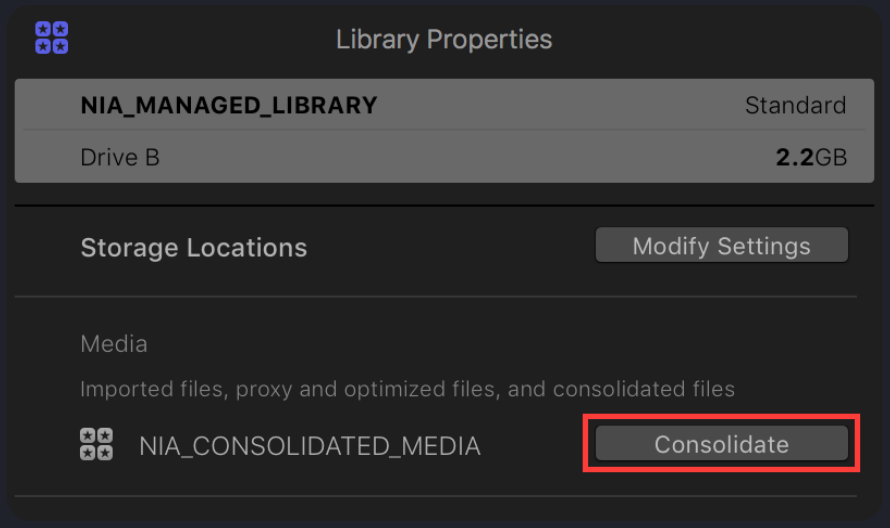



Under the Media dropdown menu in the Storage Locations dialogue box, select Choose.




Navigate to where you'd like to store your external media, creating a new folder if necessary, and click Choose. Click OK to close the Modify Settings dialogue box.

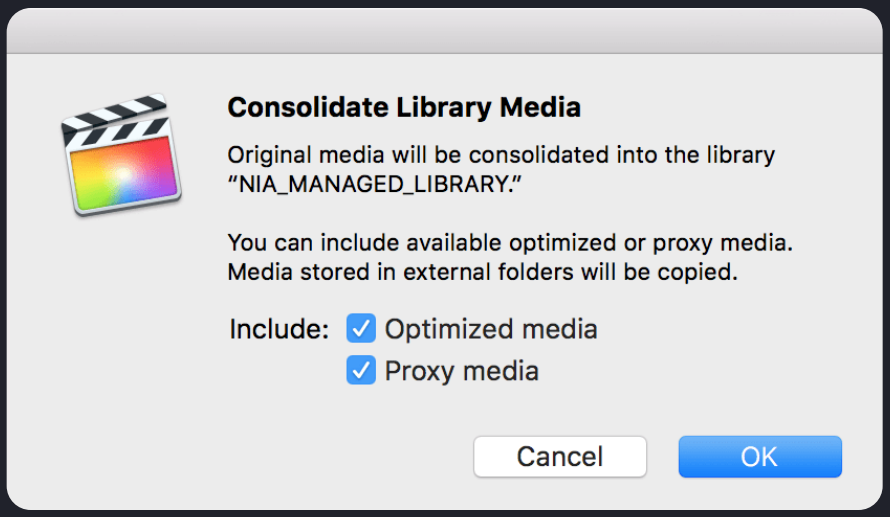
Back in Library Preferences, click the Consolidate button under the Media heading.




In the dialogue box, you'll have the option to include Optimized media, Proxy media, or both.

If you haven't generated any media then you can leave these options checked, however if you have already generated

proxy media, then uncheck the Proxy media box so it remains inside the library as managed media.

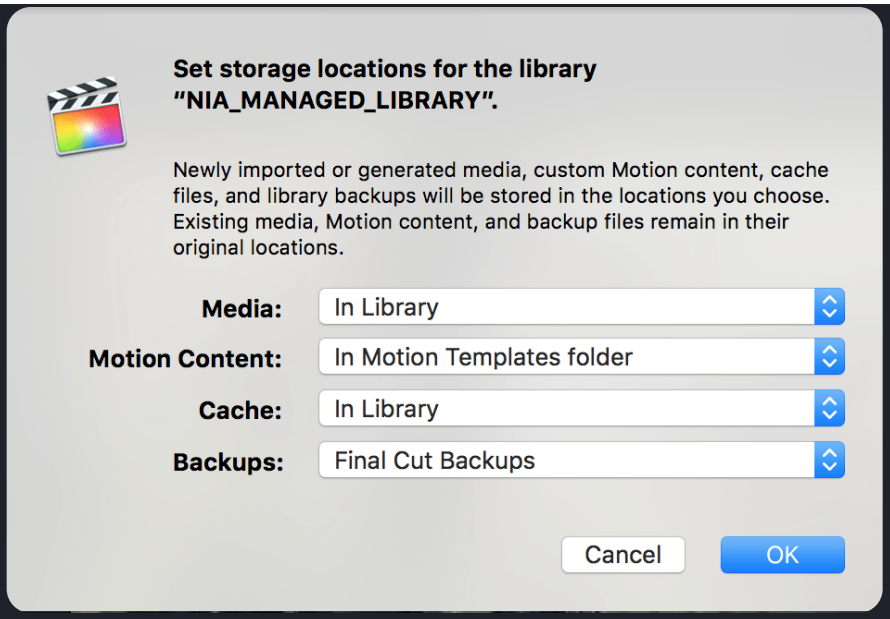



Once you click OK, FCP X will move all of your media outside of the library to the new location. If you reveal your library in the finder one more time, you can verify that your media now uses symlinks and is external.

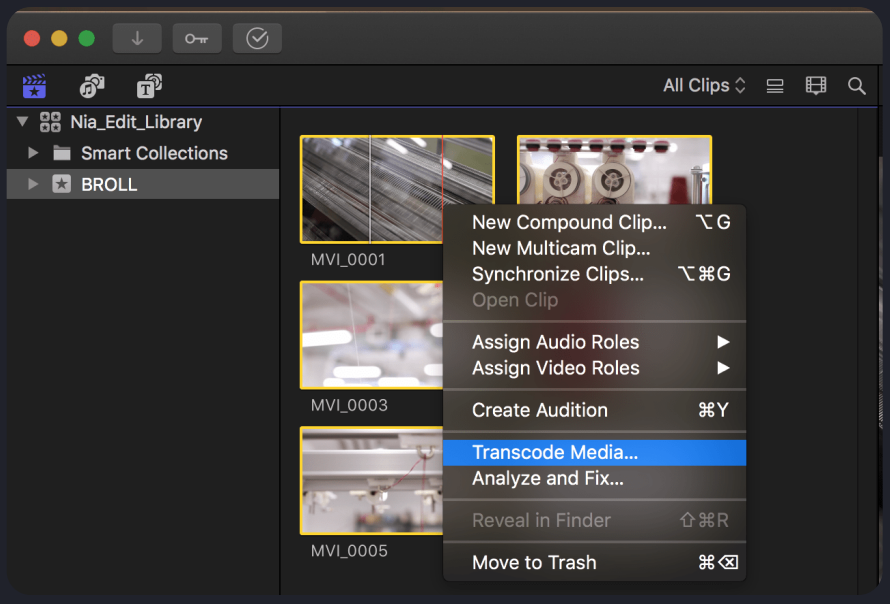
## Creating Proxies from the Browser

Before we create proxies for our clips, we need to change our storage locations one more time, so that the proxy clips end up inside the library as managed media.

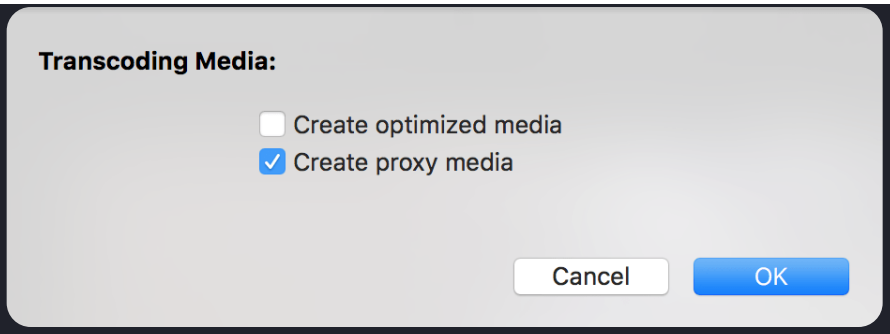
In Library Preferences, select Modify Settings next to Storage Locations. Under the Media dropdown menu, select In Library and click ok.




Now, simply select the clips or whole events you want to create proxies for, right click, and choose Transcode Media from the context menu.




The Transcoding Media dialogue box opens, with the same two choices we explored when importing: Create optimized media or Create proxy media.



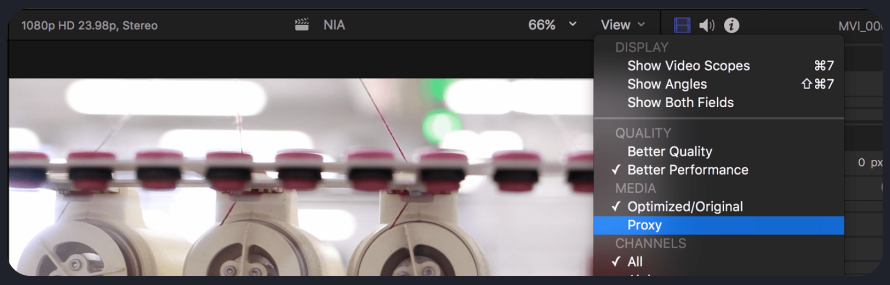

Check Create proxy media and click OK. FCP X will begin to generate proxies in the background, which you can monitor through the Background Tasks window.

## Working with Proxies in the Project

Now that we've created our proxies, we have to set the viewer to display them instead of our original camera media.

## Toggling Proxy Media

In the upper-right corner of the viewer, click the View dropdown menu, locate the MEDIA heading and select Proxy.




FCP X will automatically switch your project to view proxy media. It's important to understand that ONLY proxy media will be displayed when Proxy is selected in the View menu.

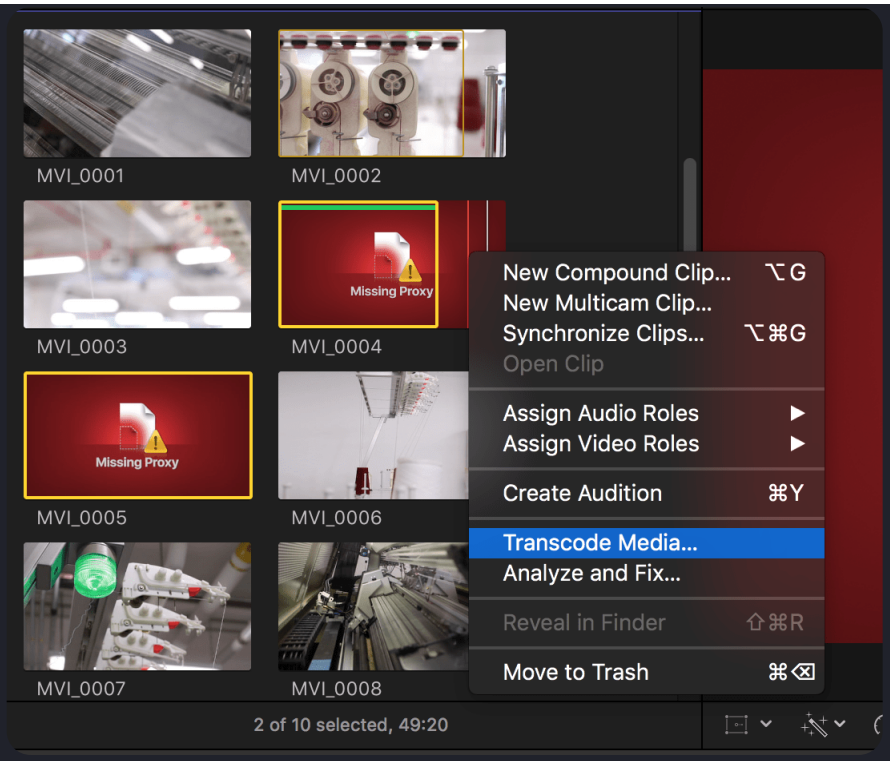
That means any files that haven't been transcoded to proxy

media will display a Missing Proxy File icon.

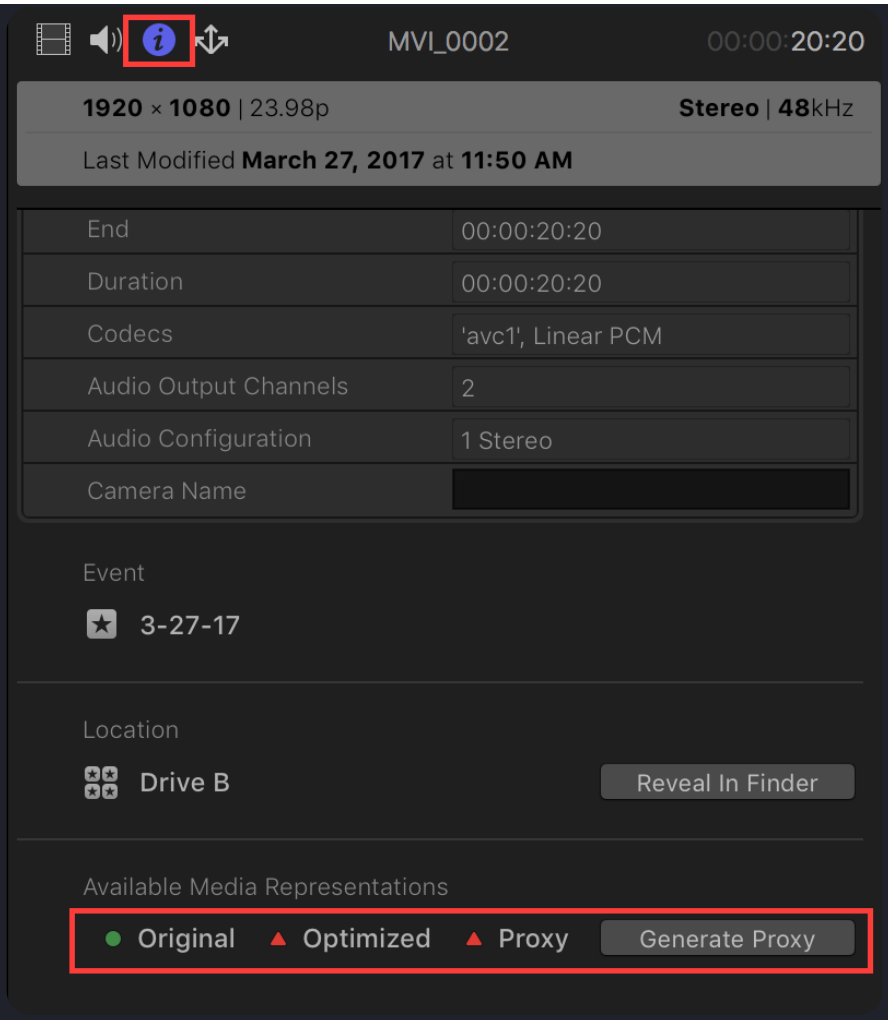
## Create Missing Proxies

If you see any Missing Proxy File icons, you can transcode the missing media in one of two ways:

- Select the clip or clips in the event browser, identified by the Missing Proxy warning, right click, and choose Transcode Media from the context menu.

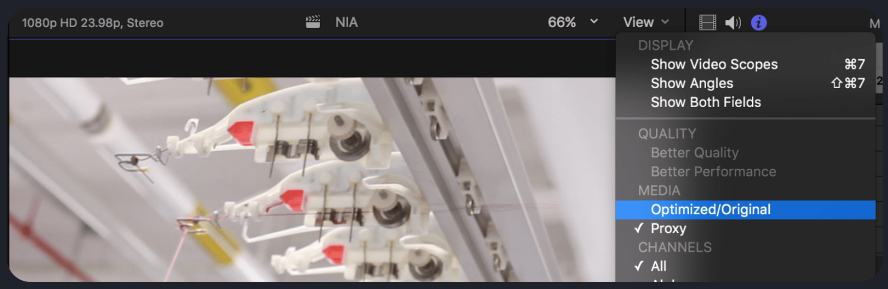



- Select a clip with a missing proxy in the event browser and open the inspector by using the keyboard shortcut "⌘4". In the inspector, navigate to the Info tab. Under the Available Media Representation heading, you'll see that the Proxy media is missing, indicated by a red triangle (a green dot means the proxy is online). Click on the Generate Proxy button to create the missing proxy.




## Exporting Your Project

When you're ready to export your project for sharing or review, you'll almost always want to switch back to Optimized/Original media in the viewer by clicking the View dropdown menu and selecting Optimized/Original under the Media heading.

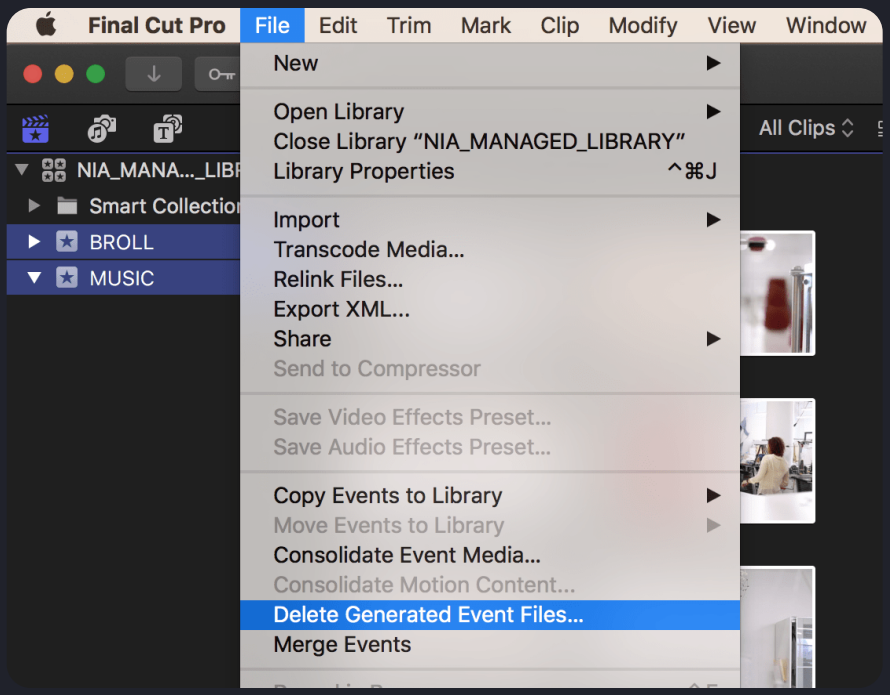


By default, FCP X will export using whichever setting is active. For the highest quality export, always use Optimized/Original media. However, if you'd prefer to export a smaller, lower quality file for review, then exporting proxy media can be desirable as well, as it will usually be faster.

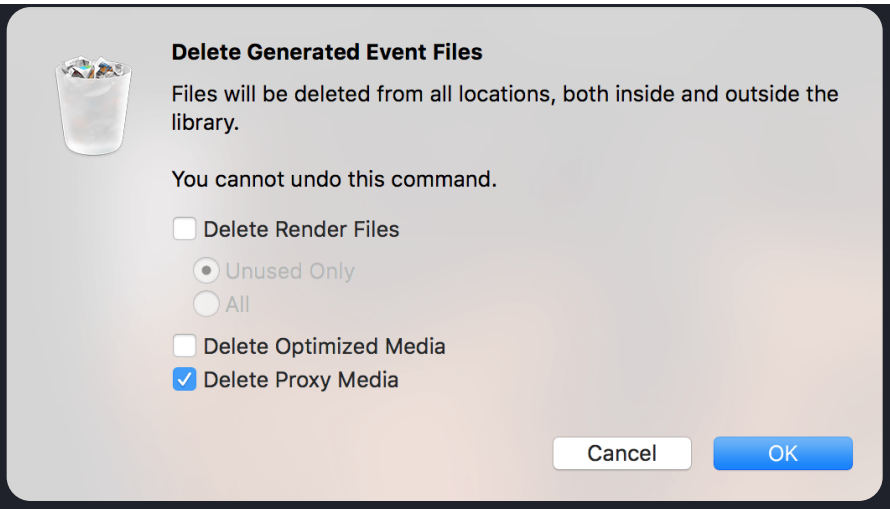
## Deleting Proxy Media

When you've finished a project and don't need the generated proxy media anymore, you can easily delete it from within FCP X.

Select the event or events you'd like to delete proxies for and navigate to File > Delete Generated Event Files.




Check Delete Proxy Media and choose any other options you'd like in the dialogue box. Click OK and FCP X will delete the proxy media.




## Copying Proxy Media to a Portable Drive

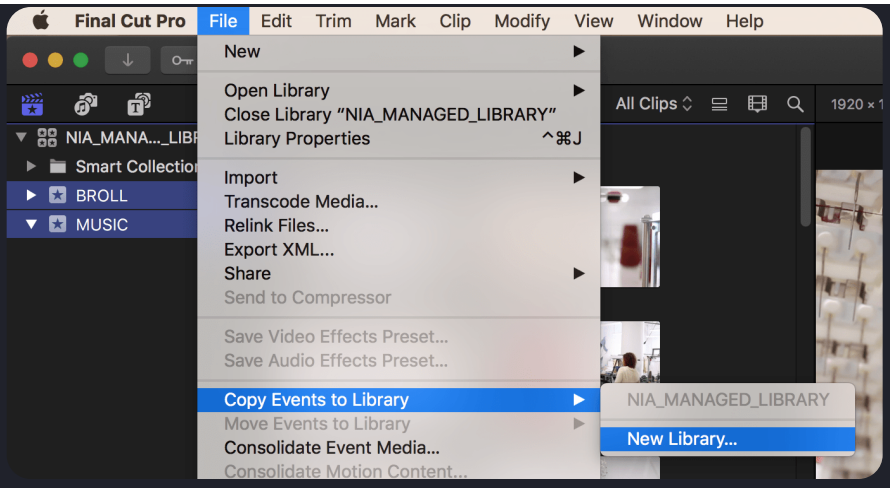
Now that we have camera media in an external folder and proxy media managed in the library, moving just our proxies to a portable hard is simple. We can go about it into two different ways.

But first, take a moment to open your project and change the Viewer to display proxy media. If anything in your timeline is offline, create the missing proxy media now as detailed above.

If everything is online, you're ready to copy.

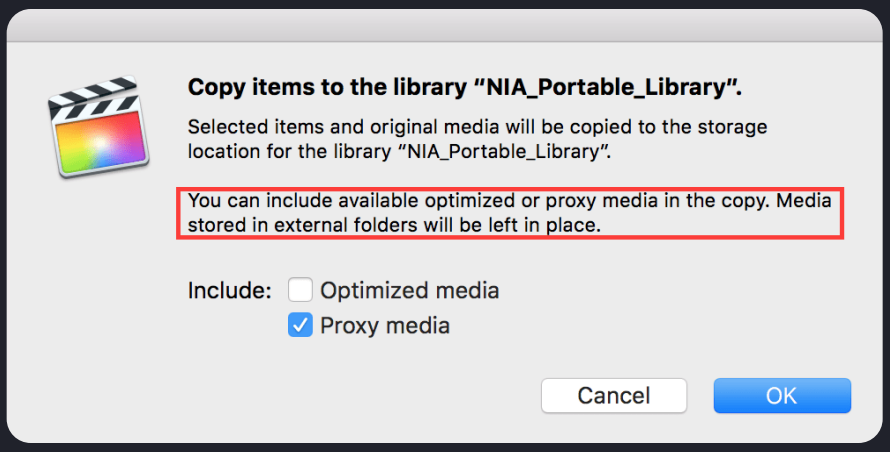
## Copying from Within FCP X

In the sidebar in FCP X, select all the events in your library. Navigate to File > Copy Events to Library and select New Library. In the Save dialogue box, navigate to your portable drive and create a folder if desired to house the library.




Name the new library carefully. You always want to be able to differentiate your current library from older versions.

Once you click Save, a dialogue box will pop up asking if you want to include Optimized or Proxy media. Make sure that Proxy media is checked, since we want to copy that over to our portable drive.




Notice that the information in the dialogue box tells us "Media stored in external folders will be left in place." That means that while we can tell FCP X to copy our proxy media, it will leave our original camera files in place in their external location, which is exactly what we want.

Once you click OK, FCP X will copy the library and proxy media to your portable drive.

## Copying from the Finder

We can also copy our proxy media through the finder. In the sidebar in FCP X, select your library, right click and choose Reveal in Finder from the context menu.

By now, we know that the revealed library file contains our proxy media and project metadata, but not our original camera media.

That means we can simply copy this library file over to our portable drive and all the proxy media will go along with it. We can then open it from our portable drive and get to work.

## Don't Forget to Copy External Assets

Whether you copy your library through FCP X or the finder, it's important to remember to copy over any additional media you may need on the road that's not managed within the library.

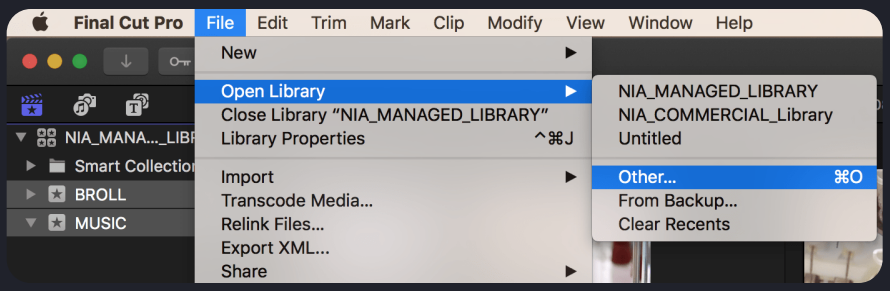
Depending on how you like to work and import media, this may include an external music, graphics or ADR folder.

Remember that when you move this type of media between drives and reopen your library, you'll have to relink it through FCP X's Relink Media window.

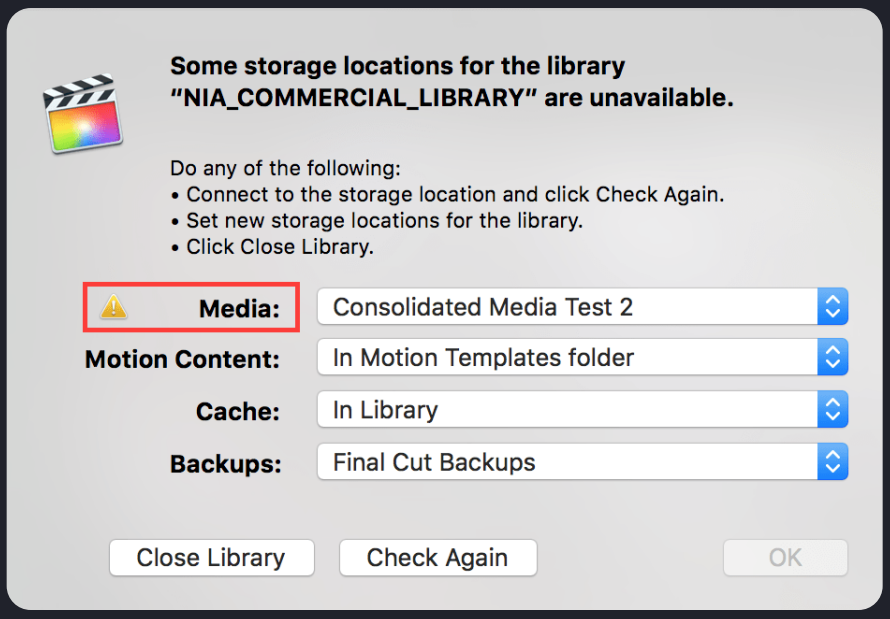
## Open the Library from Your Portable Drive

Once your library is copied over to your portable drive, you

can open it either by double clicking it in the Finder window or from within FCP X by navigating to File > Open Library > Other and then choosing the library file on your portable drive.




When the library opens, you may get a warning that some storage locations for the library are unavailable. You'll see a caution triangle next to any locations that need to be addressed, but in most cases, it's just Media. Choose In Library from the dropdown menu and click OK.




Don't panic if your media shows up offline. Go to the View menu dropdown and make sure that Proxy is selected. Now your proxy media should be online.

Lastly, reconnect any external media, such as music or graphics files, using the File > Relink Files feature.

## Keeping Track of Updated Library and/or Project Files

How you choose to manage your library and project files is up to you, but here a few suggestions:

- If you copy your library between hard drives, append each copy with the date or a version number so you always know which is the most recent file.
- If your library becomes large with proxy media, you don't necessarily have to re-copy it every time you update your project. Instead, you can use FCP X's Copy or Move Items feature to move new events between libraries, detailed here. Just remember that if you've added new media to your library, you'll want to copy those over too including the proxy files when prompted within FCP X and manually copying the original camera files to your external camera media folder.
- There are third-party apps that help make syncing library, events, and projects easier. Which one to use really depends on your personal workflow, but two worth a mention are Sync Folders Pro and Final Cut Library Manager.

## Other Workflow Ideas

If you want to explore more and like troubleshooting workflows, then we encourage you to test these ideas out (use with caution).

## External Proxies

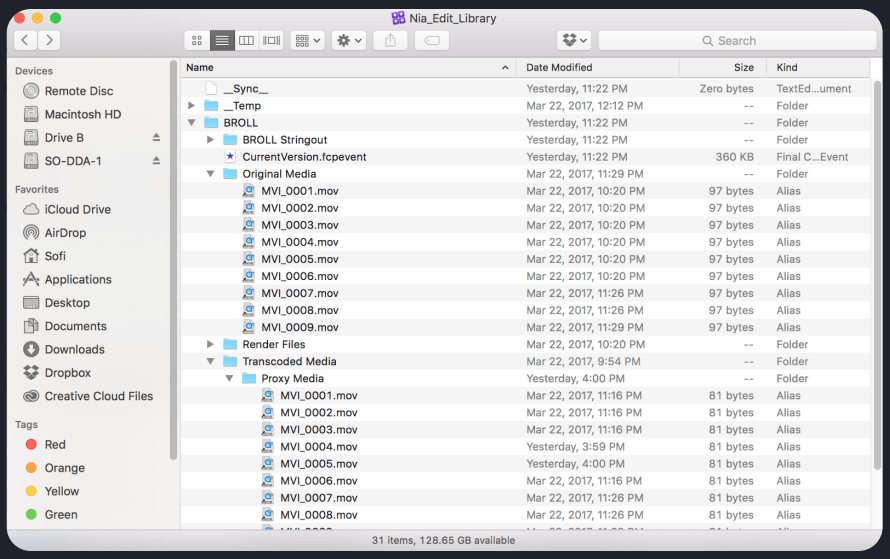
So far, we've made a clear case for keeping proxies as managed media within the FCP X Library. This is a sure-fire way to make sure that you always have the media you need and that nothing goes offline.

But sometimes for larger projects, your proxy media ends up being pretty big, which in turn makes copying your library file between drives a bit of a pain.

If you ONLY intend to use proxies on the same, single portable drive (or internally on your mac) and are certain you won't need to copy them elsewhere, then using external proxies deserves a mention.

Before creating your proxies, modify the settings for your Storage Locations in the inspector panel to set an external folder for your media. Then create your proxies as normal.

Your library will look like something like this.




To work with proxies only, simply disconnect the drive with your original camera media and take the drive or computer with your proxy media with you.

The benefit to this workflow is that because ALL the media is

stored externally, the library file itself is very small and can be copied, shared, and updated more quickly.

The downside is that if you move the file path of your external proxies or even rename the drive they're located on, FCP X's symlinks will break and the proxies will go offline with no option to reconnect then.

## External Proxies with Symlinks

Here's another intriguing workflow for dealing with larger proxy libraries that uses self-made symlinks, for those who want to experiment.

Essentially, you create the proxies as managed media and then manually move them outside the library as external media and create your own symlinks though the Finder.

One big drawback for this method is that FCP X separates each event into its own folder, which means if you're working on a larger project with many events, this workaround becomes tedious to maintain.

## Using Original Media to "Relink" Proxies

If you do find yourself in a situation where your proxies have gone offline and can't be relinked, there is one last resort that generally yields success, but requires having access to both the proxy files AND the original media.

Make sure that all media, proxy and original, is available on the computer you're working with. Select your library and in Library Properties, select the Modify Settings button for

Storage Locations.

Set your Media storage location to where the existing proxies are. Depending on how you created them, the folder structure will differ, but generally, you want to point to the outermost folder that FCP X created.

Back in the browser, select the clips that need to be relinked to their proxies, right click and choose Transcode Media from the context menu. Choose Create proxy media in the dialogue box and click OK.

FCP X should then recognize that those proxies are already there and rather than re-transcode then, simply reconnect them.

Note: In general, this method seems to work, but some people have reported that FCP X duplicates certain proxy clips.

## In-Camera and Third-Party Proxies

FCP X does not have the capacity to link original high resolution media to in-camera proxies or proxies created in other software, such as DaVinci Resolve.

You can, of course, import these proxies, edit your offline project and then relink back to the original media to create an online project. But there is no way to take advantage of FCP X's fast switching between original and proxy media with this method.

## Wrapping Up

And the more you understand about how FCP X works internally, the more you can adapt any given workflow to meet your needs. Using proxies in the most basic situation requires nothing more than checking one box, but if you want to become a proxy master, it's worth putting in the time to really understand the various options. FCP X's tools get the job done, combining your offline and online edits into one. So create a library, import some media, and start playing around.

## Premiere Pro Proxy Workflows

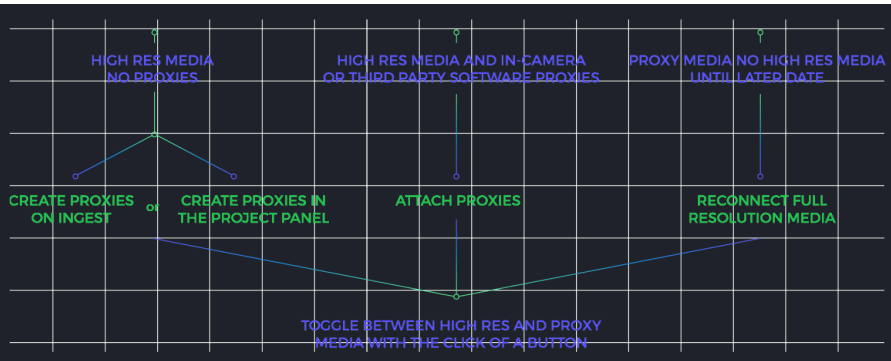
Premiere Pro makes proxies easy. Really easy. Like, toggle back and forth with the click of a button, easy.

So put away those thoughts of an arduous, error-prone process and embrace Premiere Pro's integrated, simple, and intuitive proxy workflow. With a little know-how, anybody can take advantage of its benefits.

By the end of this article, you'll have a complete understanding of Premiere Pro's Create Proxies, Attach Proxies, and Reconnect Full Resolution Media workflows.

This simple chart can help you determine where to begin and which workflow is most relevant.

It all depends on your starting media. If you have…




If you can identify your workflow, feel free to jump ahead to a specific section, or read on to gain a complete understanding of Premiere Pro's proxy workflow.

## What Are Proxies Again?

Proxies are lower resolution copies of your media that are created in one of two ways – either by transcoding raw media to an intermediate codec or by simultaneous in-camera recording, a standard feature on many high end cameras.

Editing with proxy media is commonly referred to as an offline edit, while the process of relinking back to the full res media at the end of a project is called the online edit. Premiere Pro's proxy workflow essentially combines the two, allowing you to effortlessly switch between offline and online, all within the same software.

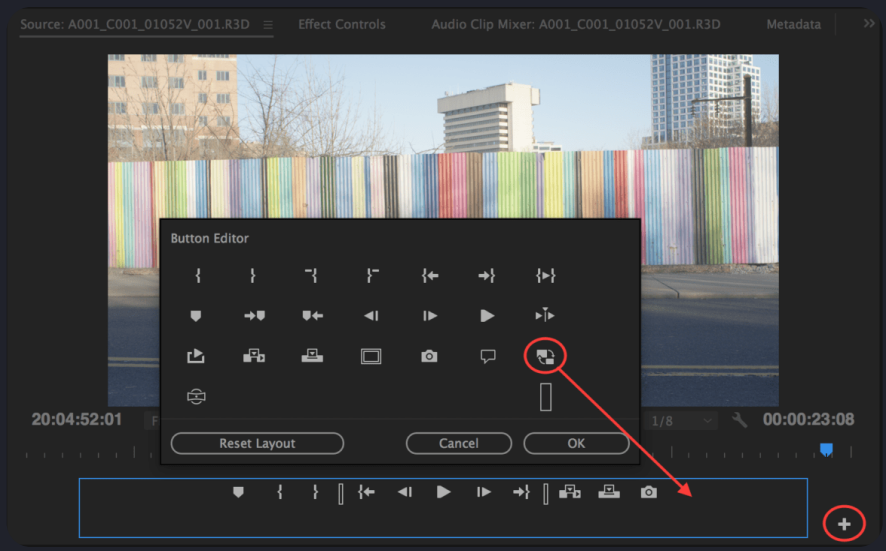
So, how does proxy editing work? Let's start by adding a simple button.

## Adding the Toggle Proxies Button

The Toggle Proxies Button is powerful. With one click, you can seamlessly switch back and forth between proxy and full res

media. By default, it's not mapped to the monitors or the keyboard, so let's add it.

In the bottom right of either the Source Monitor, Program Monitor, or both if you'd like, click the Button Editor (+). This opens up a panel of buttons we can add to the transport control area of the our monitors. Hover over a button icon to reveal its name, or refer to the image below for the Toggle Proxies Button icon.




Click and drag the icon in line with the other buttons in the transport control area. When the button clicks blue, you're viewing proxies in the Source and Program Monitors. When it's white, you're looking at your full resolution media.

You can also set a shortcut for Toggle Proxies in Keyboard Shortcuts, if you'd like, or alternatively, you can find the same setting in Preferences –> Media –> Enable proxies. The buttons, keyboard shortcut, and preference are all linked, so setting it in one place sets it globally.

Now that we know how to toggle proxies on and off, let's get started creating them.

# 1. Creating Proxies in Premiere Pro

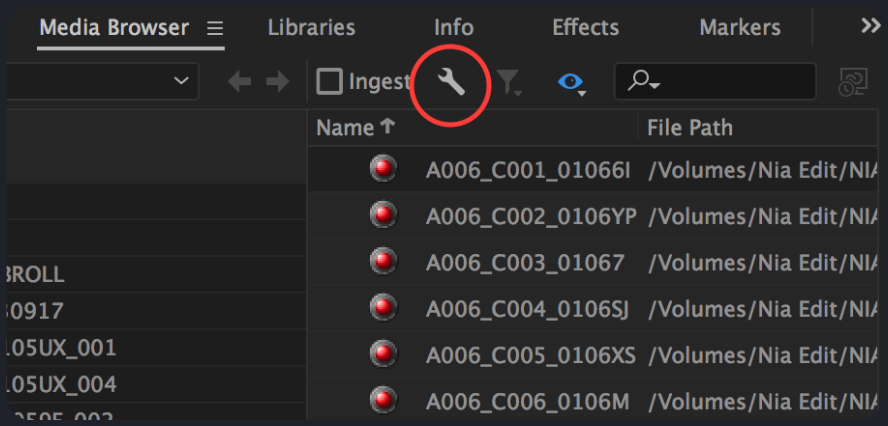
## Understanding Ingest Settings

Most of us know a few different ways to import footage into Premiere Pro. You can choose "Import" from the file menu, double click empty space in the project panel, use the media browser to locate and import clips, etc.

By default, when we import footage, we simply create a clip in the project panel that points back to the files on our hard drives. The original media isn't changed, duplicated, or moved in order for us to work with it in the project.

By adjusting our Ingest Settings, we can tell Premiere Pro to behave differently when we import media.

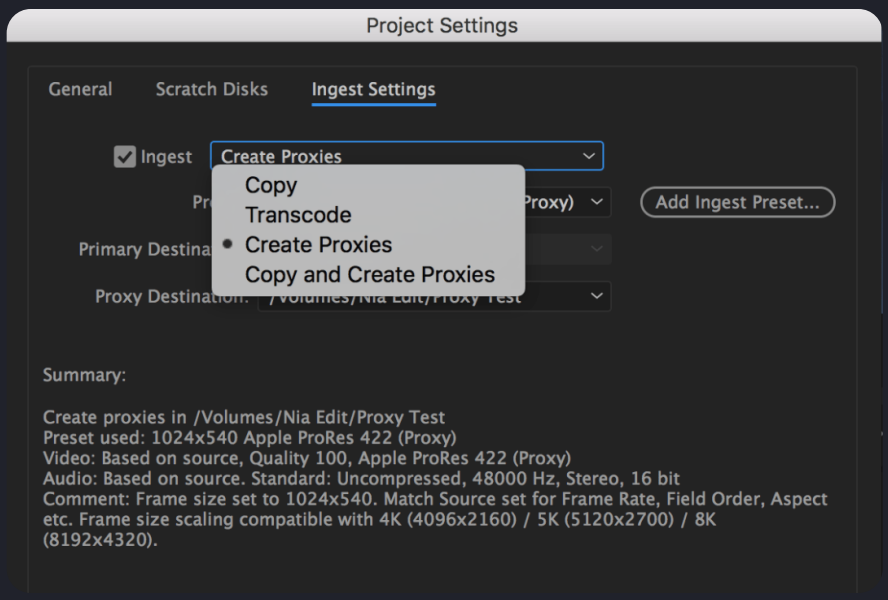
To open Ingest Settings, navigate to the Media Browser and locate the "Ingest" checkbox on the top right menu bar. Click the wrench to the right of the Ingest check box to open Ingest Settings. Alternatively, go to File –> Project Settings –> Ingest Settings.




By default, Ingest is unchecked and our options are greyed out. Checking the box here will also check it back in your

Media Browser, meaning that the settings you choose will be applied on your next import unless you uncheck the box.

When checked, four ingest options are available in the drop down menu. We're mostly concerned with the last two, but let's briefly go over each one.




- Copy and Create Proxies: Copy camera media as above, and create and attach proxies for that media as above.

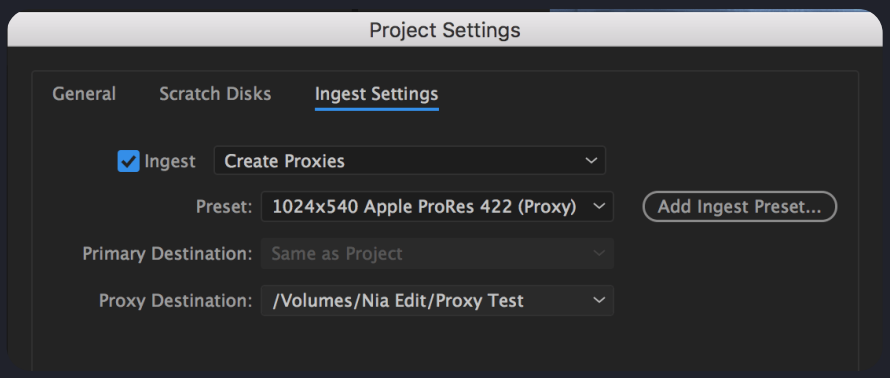
## Creating Proxies on Ingest

Let's choose the Create Proxies option and take a closer look, noting that Copy and Create Proxies is identical in function, with the added step of setting the Primary Destination for your copied source media.

Create Proxies is ideal when you're working with large files, such as 5K, 6K or 8K media, with no in-camera or third party software proxies (such as proxies created in DaVinci Resolve).

Once we've selected Create Proxies, we have two additional

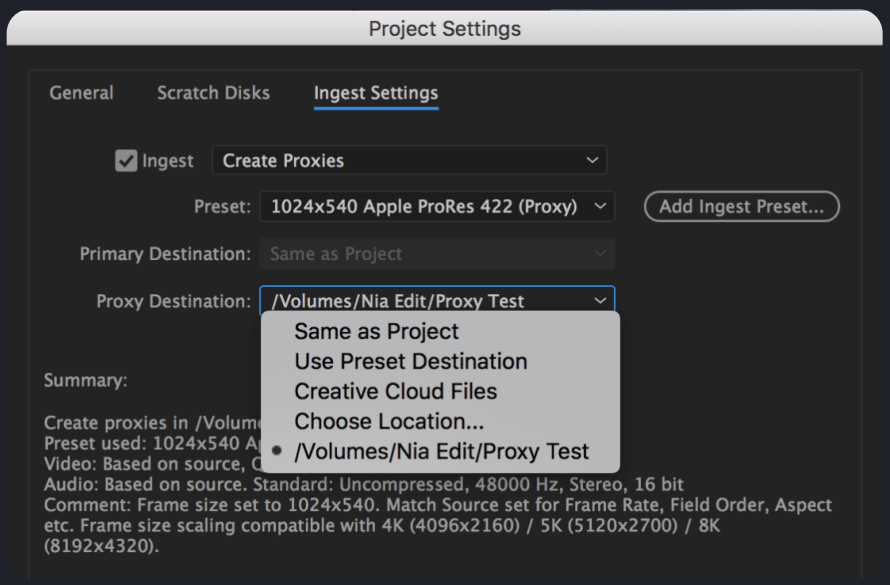
options: Preset and Proxy Destination.




Let's go over Proxy Destination first.

## Choosing a Proxy Destination

In the drop down menu, you can determine where Premiere puts the generated proxies.



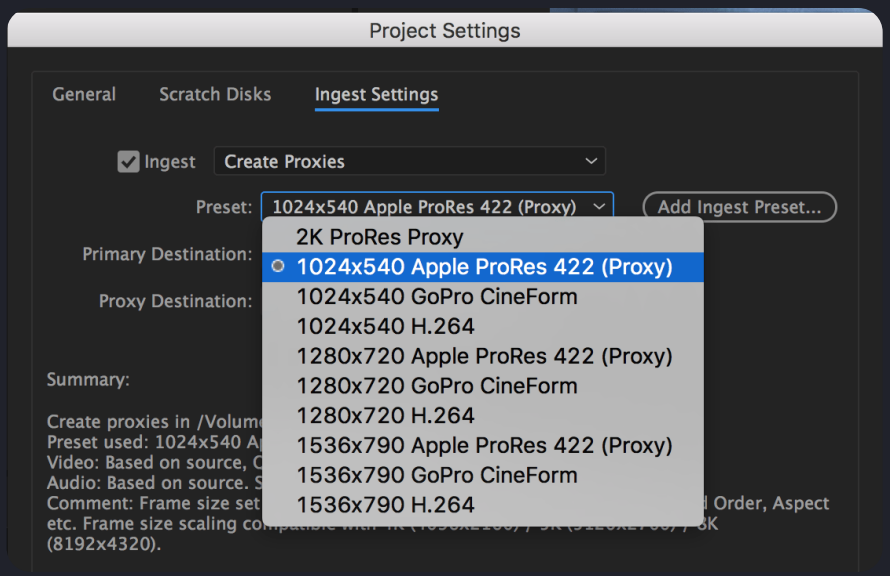

I advocate for selecting Choose Location and creating a dedicated proxy folder on your hard drive. That way, you'll have all of your files in one place, which means you can easily copy them to portable hard drives for easy sharing or for travel.

The Creative Cloud Files option deserves a mention as well. This choice automatically syncs your proxy files to your Creative Cloud Files Folder, which in theory, means you have access to your proxies wherever you're able to connect to your Creative Cloud account.

In reality, remember that you'll need a significant amount of cloud storage to accommodate your files, as well as a fast internet connection to sync them across different devices.

## Choosing a Preset

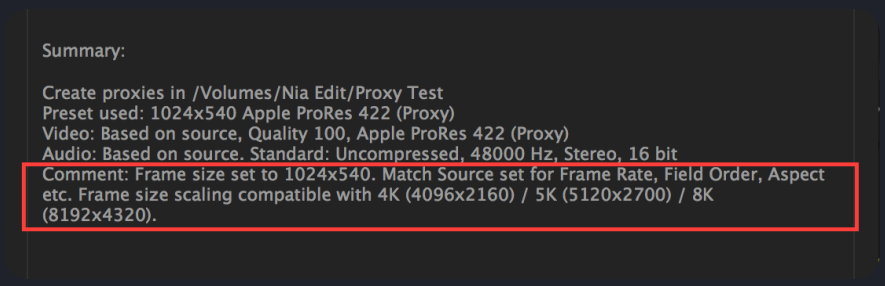
If you click the Preset dropdown menu, you'll see that Premiere Pro has a list of defaults. While not comprehensive, these are usually good options to consider. If they don't appeal to you, we'll cover creating your own presets a little further on.




The Preset that you select will determine the codec and frame size of the proxy files. The beauty of a proxy workflow is that you always have the option to toggle back to your full res media. That means we can choose a preset that's optimized

for high editing speeds and low disk space.

You'll notice that Premiere Pro has also preselected a few frame sizes for you. If you choose a preset, let's say 1024×540 Apple ProRes 422 (Proxy), you can look in the Summary section below for suggestions of compatible source frame sizes.




We can see that this setting works well with 4K, 5K, and 8K media because the frame dimensions share the same aspect ratio. Matching your proxy aspect ratio to your full res media is always preferable and yields the best results.

## What About HD Frame Sizes?

It's worth noting here that frame sizes with different aspect ratios, though not ideal, can work too.

Many people prefer to do their offline edit in the standard HD sizes of either 1920×1080 or 1280×720. If you've worked with 4K, 5K, or 8K footage before, then you know that reducing these resolutions to fit into an HD frame creates letterboxing, which means you'll have thin black bars on the top and bottom of the frame to compensate for the slight difference in aspect ratio.

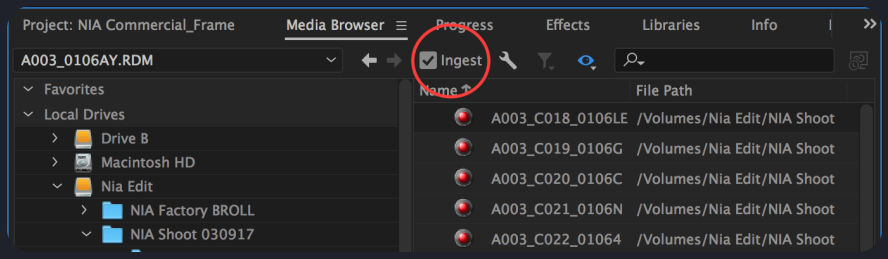
I've found that Premiere can handle proxies with letterboxing, but we don't recommend it for the following reasons:

- When you toggle between your proxy and high res media, you'll see your letterboxing appear and disappear.
- If you add titles or any effects that affect only part of the image (say, a masked color correction), they may not line up as intended when you switch back to the original files.
- Premiere always uses the full res clip while rendering, so if you partially render a clip, when you play it back, your letterboxing will disappear for the rendered portion of the clip, and reappear for the unrendered portion.

## Background Processing and Monitoring Progress

Once you've selected a Preset and a Proxy Destination, it's time to let Premiere Pro go to work creating your new files. Premiere Pro's proxy workflow is seamlessly integrated with Adobe Media Encoder, which is where the encoding will actually take place.

As long as the Ingest box is checked in your Media Browser panel, Premiere will automatically add files you import that are eligible for proxies (audio files, graphics, etc., will not be impacted) to Media Encoder's queue, applying the presets you selected in Ingest Settings.

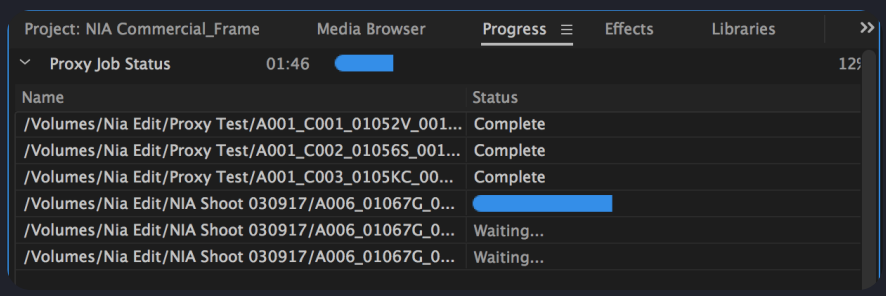



It doesn't matter how you choose to import, whether through the Media Browser, dragging and dropping into the Project Panel, etc., Premiere will apply your Ingest Settings and send a job over to Media Encoder to create your proxies.

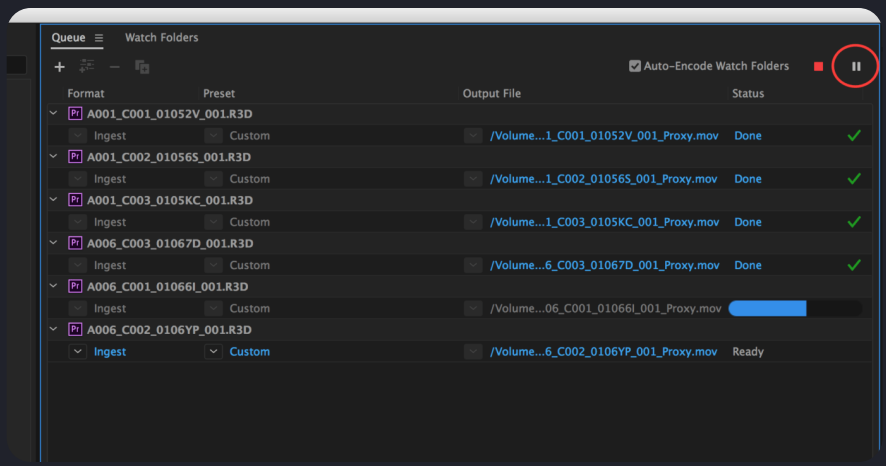
A great feature here is that Premiere and Media Encoder use background processing. This means you can keep working in Premiere, labeling or even editing your clips, while the proxies generate in Media Encoder. When they're done, they'll automatically attach to your high res clips in the Project Panel.

You can monitor the background processing in two ways:

- In Premiere: Navigate to Window –> Progress to open the Progress Panel. You can leave it as a standalone panel, or dock it by dragging it near the Project Panel and letting go.




The Progress Panel shows you a list of all the proxies currently being generated, along with a progress bar. When they're finished, the progress bar will change to Complete. In Media Encoder: You'll notice that a few moments after importing clips, the Media Encoder application will open up in the background and get to work. The encoding clips will be added to the queue like any other job and you can monitor progress there as well.



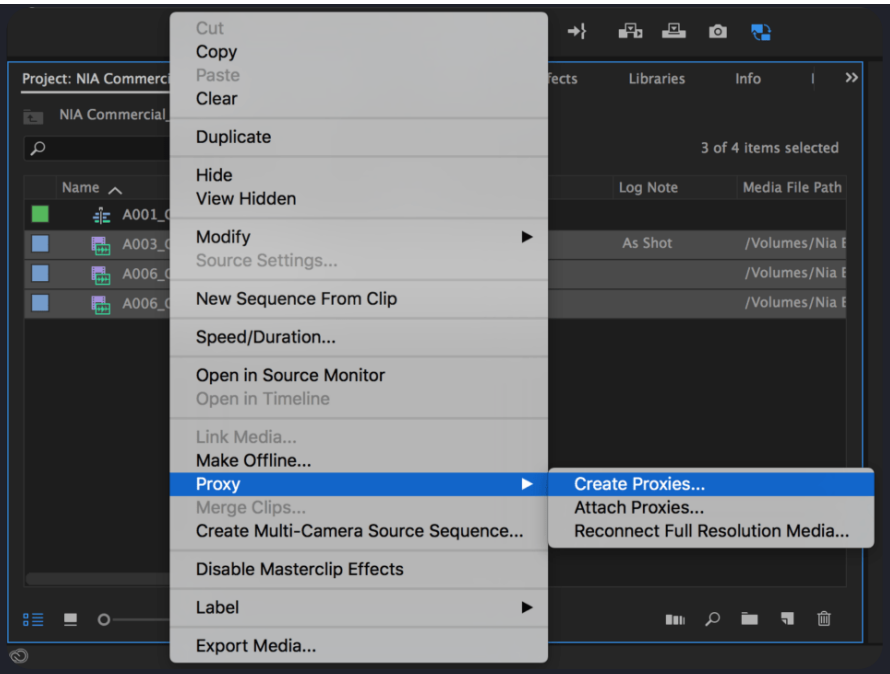
While in Media Encoder, you can you can pause and restart the render queue without disrupting its connection to Premiere. However, if you stop the render queue in Media Encoder, the connection between the proxies and original files will be lost and you'll have to connect them manually (see the Attaching Proxies section below).

## Creating Proxies from the Project Panel

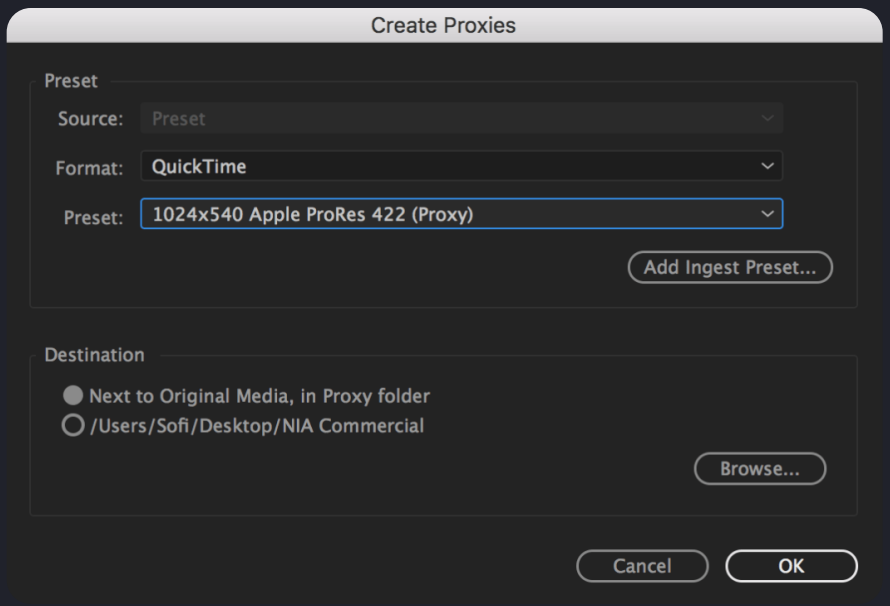
If you've already begun editing or importing without having created proxies on Ingest, don't worry. You can still create them, right from the Project Panel, and have Premiere automatically attach them to your high res media.

I sometimes use this option when your imported camera media isn't performing as well in Premiere as I'd like it to. Certain codecs perform better than others (here's an excellent explanation for that) and if your editing speed is really suffering, you may choose to create proxies mid-project to complete the edit.

In the Project Panel, select the media that you want to create proxies for and right click on it. In the context menu, choose Proxy–> Create Proxies.




The Create Proxies dialogue box appears. While it looks slightly different than Ingest Settings, it contains the same options and information.

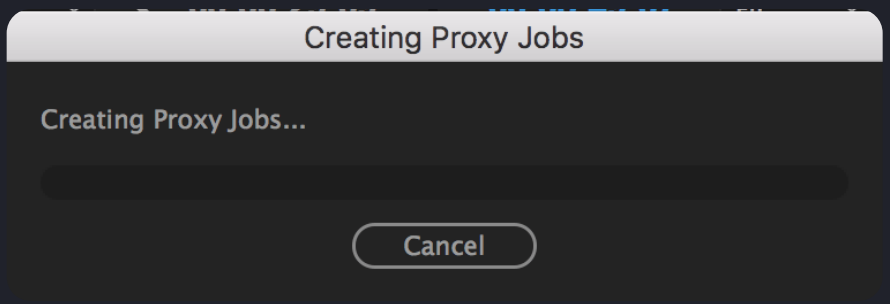



If you click on the Format drop down, you'll see two default options: H.264 and Quicktime. These specify the general categories that your proxy presets fall under. Select each one and you'll see that the options under the Preset menu change, and are the same choices we have in Ingest Settings.

There are two options for Destination:

- Next to Original Media, in Proxy Folder: If you choose this option, Premiere will create a new folder for each proxy clip, in the same directory as the original high res media. Because most high end cameras have a complicated file structure where each clip is isolated in its own folder, this option generally isn't ideal. You'll have a hard time grouping the proxies to move them to another drive, or deleting them when you're done with the project.
- Browse: you almost always suggest browsing and selecting your own destination to ensure that all of the proxies are in the same folder.

Once you set your Preset and Destination and click OK, a Creating Proxy Jobs progress bar pops up, which indicates that Premiere is sending the clips over to Media Encoder. Note that if you cancel this progress bar, the whole proxy creation process will be canceled and the clips will never be added Media Encoder's queue.




Once the progress bar is complete, you can track the progress of the proxy creation as detailed above, either through the Progress Panel in Premiere, or by switching over to Media Encoder.

## Creating Custom Presets in Adobe Media Encoder

In some cases, you may want to create your own preset. This option helps when you need to set a custom frame size or choose a slightly higher quality codec, which most often happens when you're traveling and will only have access to the lower resolution proxy media.

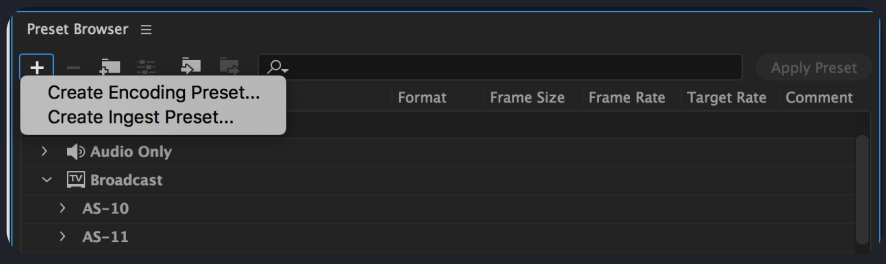
Why? Because one of the benefits of working with 5K or larger media is that we have tons of latitude for repositioning and zooming. Being able to create a close up in post has can really save you.

But if you're using the 1024×540 proxy preset, a frame that's slightly smaller than 720p HD, you may have trouble discerning if your focus is still sharp. Using a 2K frame size (2048×1080) will be more useful to you, especially if you know your laptop can handle it nearly as well, as long as you choose the right codec.

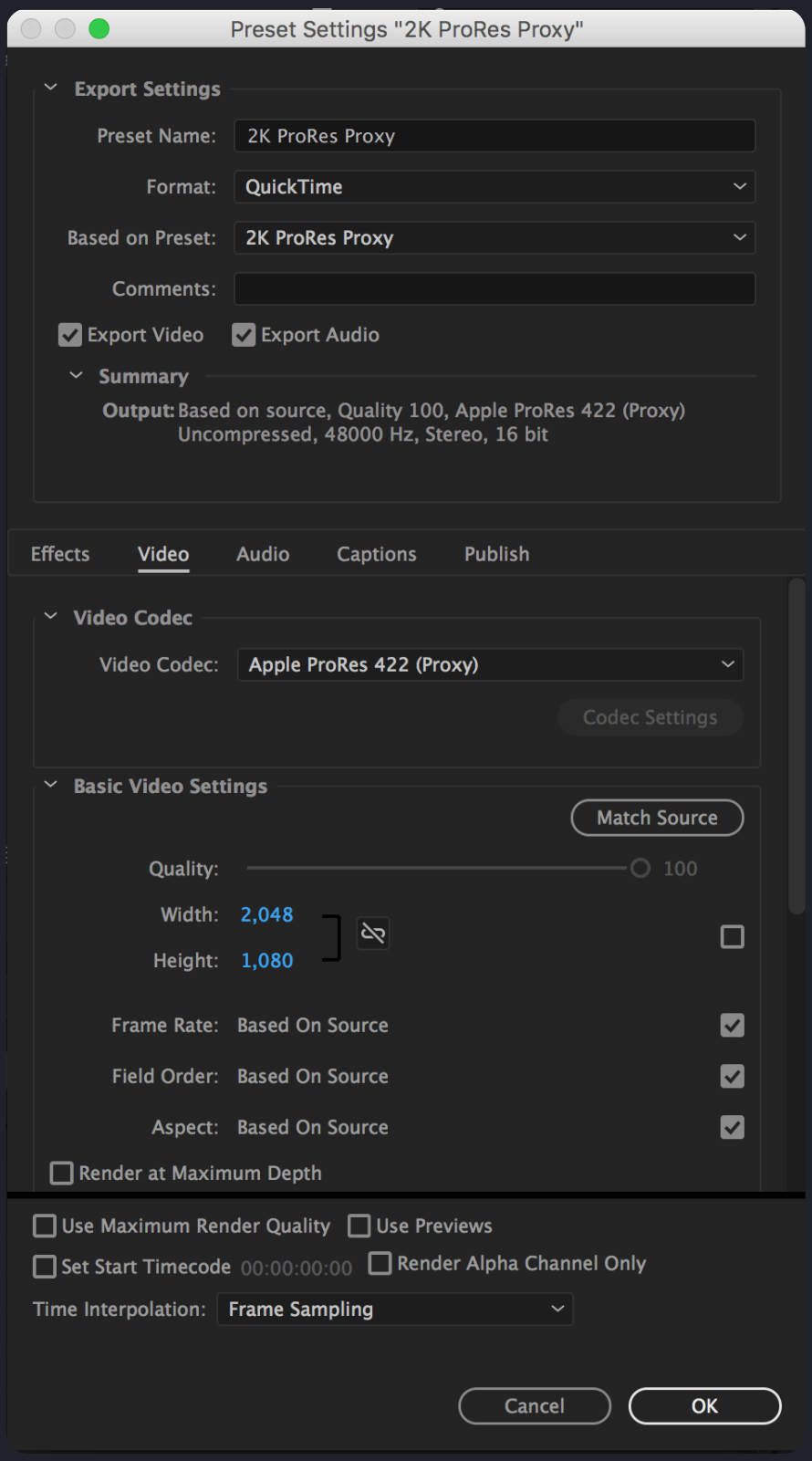
My favorite codec is a lightweight flavor of Apple ProRes, like ProRes Proxy, ProRes LT, or even ProRes 422. If you work primarily on a PC, then you should likely opt for a similar flavor of DNxHD. Premiere handles both codecs extremely well.

To use your own preset in Premiere's Ingest Settings, we need to first create it.

Open Adobe Media Encoder and navigate to the preset browser in the lower left of the screen. Click on the new preset button (+) and choose Create Encoding Preset.

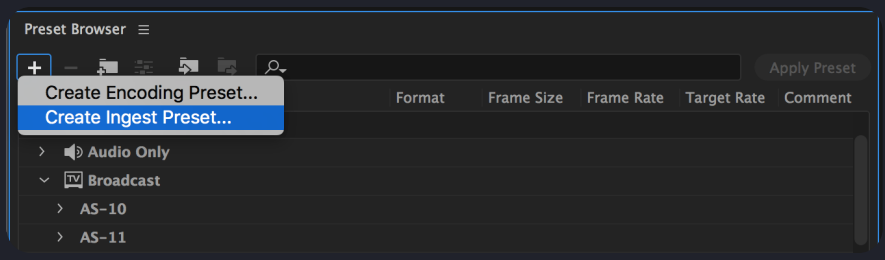


In the dialogue box, set your custom options and hit OK. Here's an example of a 2K ProRes Proxy Preset.



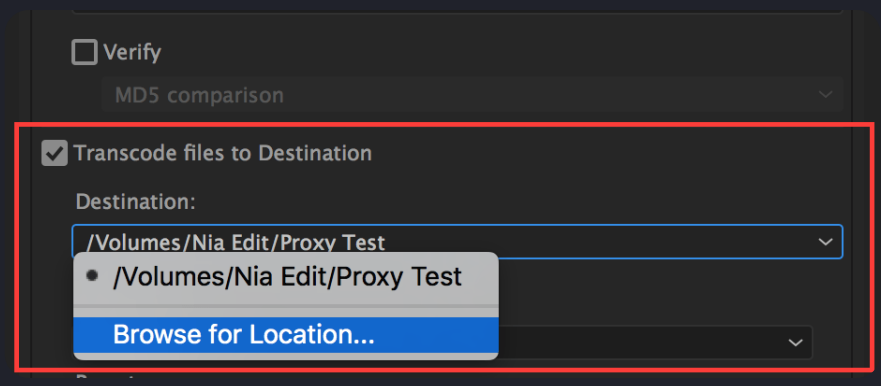

Now, back in the preset browser, click on the new preset

button (+) and choose Create Ingest Preset.

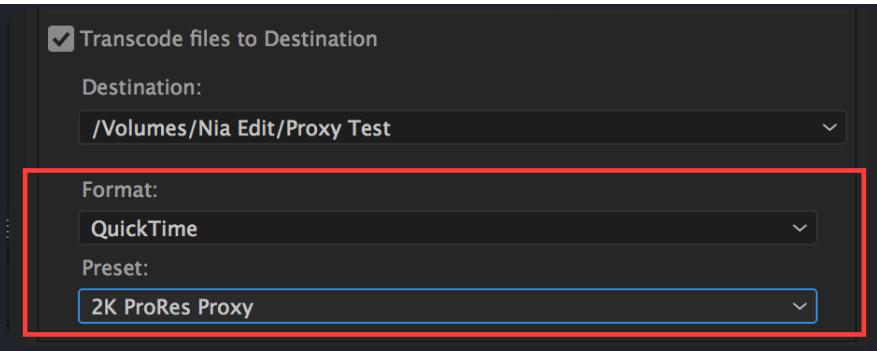



Name your Ingest Preset and then click the checkbox next to Transcode files to Destination. This option tells Premiere that the preset can be used for either the Transcode or Create Proxies Ingest Settings.*

Under the Destination drop down, click Browse for Location and choose a destination for your proxy media. You can always change the destination when choosing your Ingest Settings back in Premiere.




Under format, choose the appropriate option, which for this 2K ProRes Proxy preset is Quicktime. Once you've selected the format, you're able to choose the custom preset under the Preset dropdown.



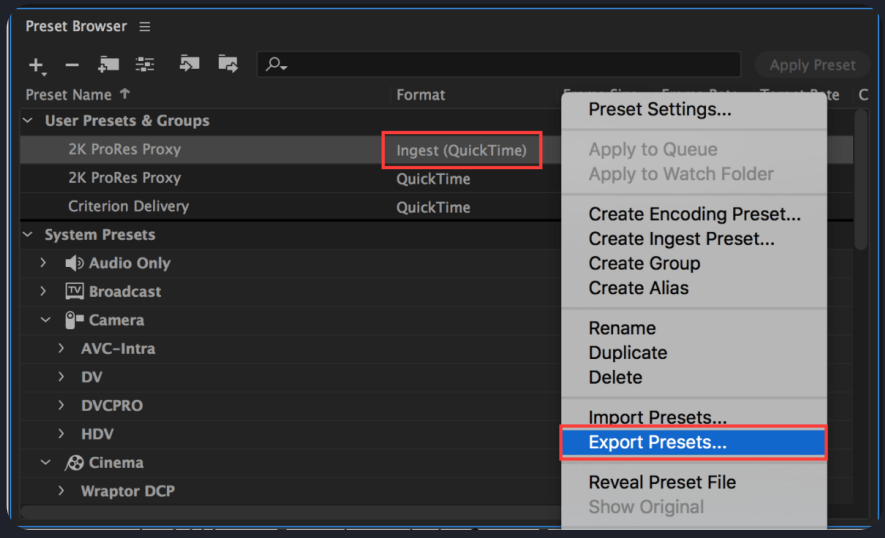

When you're finished, click OK.

## Next, export from AME

To import your newly created Ingest Preset into Premiere Pro, we first have to export the preset from Adobe Media Encoder to our hard drive.

Locate your preset in the Media Encoder Preset Browser, noting that your ingest preset will be labeled as Ingest under the Format column, while you're Encoder preset will be listed as the format you chose, such as Quicktime, H.264, etc.

Right click the preset in the Adobe Media Encoder Preset Browser, and choose Export Preset.



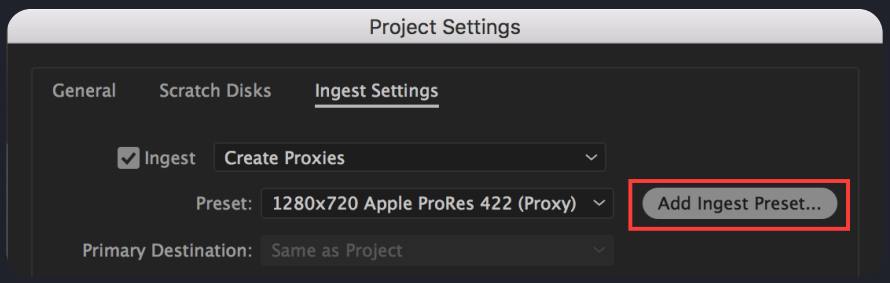
Select a location on your disk for the .epr file and click Save.

(*Note: Checking Copy files to Destination when creating an Ingest Preset will make your preset available under the Copy AND Copy and Create Proxies options in Ingest Settings. Checking both Copy files to Destination and Transcode files to Destination will make your ingest present available ONLY under the Copy and Create Proxies option.)

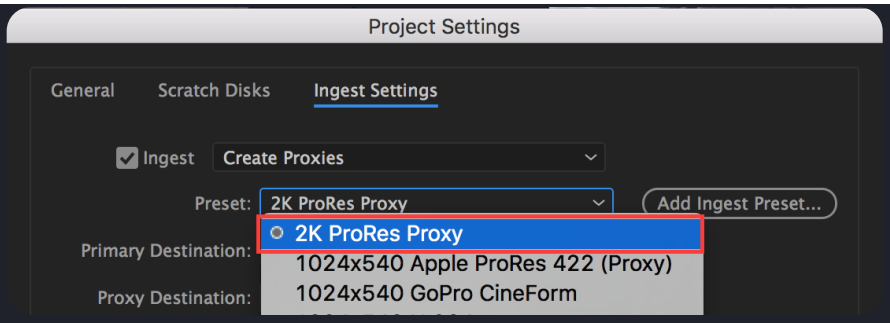
## Importing Your New Ingest Preset into Premiere

If your Ingest Settings dialogue box isn't still open in Premiere, navigate back to it, either by clicking on the wrench in the Media Browser, or going to File –> Project Settings –> Ingest Settings.

Make sure Ingest is checked with Create Proxies selected, and click the Add Ingest Preset… button next to the Preset dropdown menu.




Navigate to your saved .epr file and choose Open. Your custom setting will import into Premiere and is now a selectable option under the Preset dropdown menu.




If you're creating proxies in the Project Panel, rather than via Create Proxies on Ingest, you can use the Add Ingest Preset button in the Create Proxies dialogue box in the same way.

## 2. Attaching In-Camera Proxies and Third Party Software Proxies

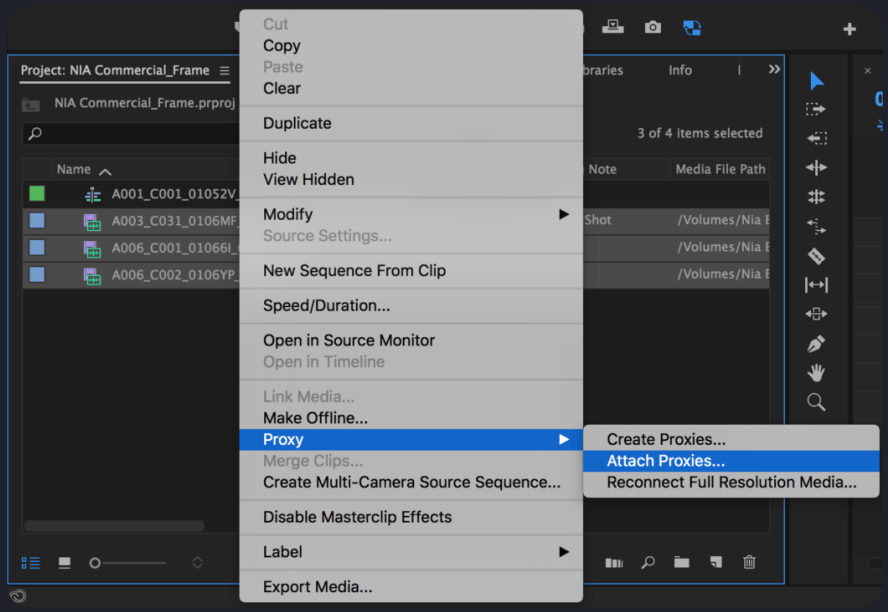
You won't always need to generate proxies on import as we've been detailing. Many high end cameras now have the ability to simultaneously record proxies right alongside the high resolution media. And for those cameras that don't have this function, there are hardware solutions that make it possible.

Another very common scenario is that a third party software is used to create proxies, like DaVinci Resolve. Many people like to use Resolve to apply LUTs or watermarks to their footage, or to otherwise take advantage of Resolve's powerful color correction tools. This is especially common on projects that shoot in LOG, which produces a very flat, desaturated image that benefits greatly from color correction, even in the early stages of the edit.

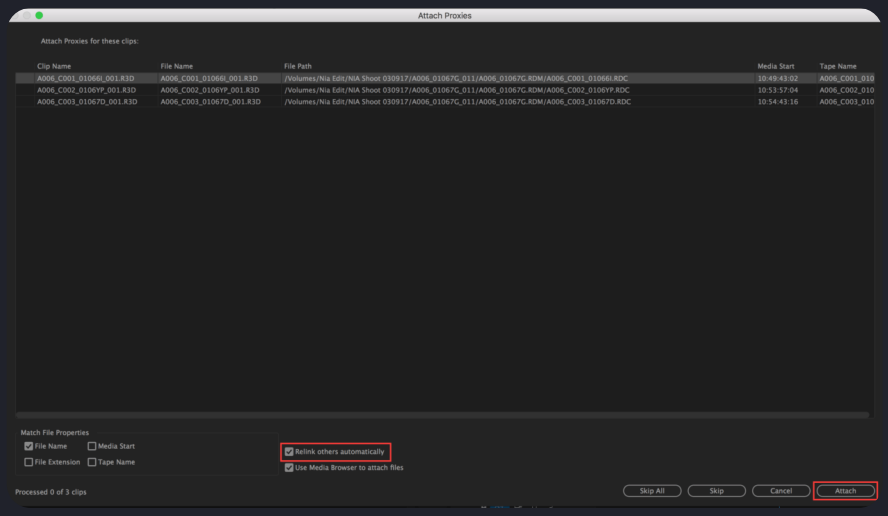
Luckily, Premiere has an Attach Proxies function that allows us to link existing proxies to our full resolution media and gain all of the same benefits as if they had been created in Premiere.

In the Project Panel, select the clips that you want to attach

proxies to and right click. In the context menu navigate to Proxy –> Attach Proxies.



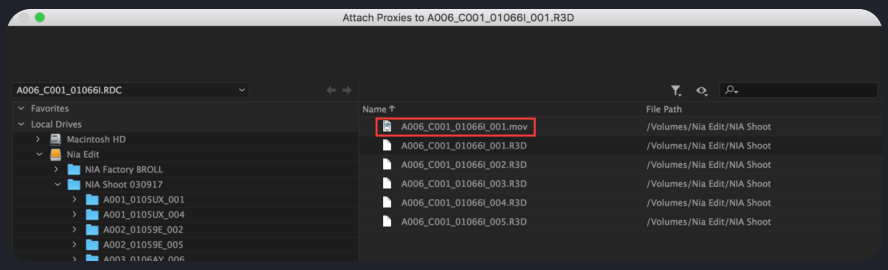

The Attach Proxies dialogue box opens, which looks nearly identical to the familiar Relink Media dialogue box and lists all the clips you've selected. Make sure the Relink others automatically box is checked and click the Attach button in the lower right corner.




The Media Browser pops up, allowing us to navigate to the corresponding proxy to attach. If you're attaching in-camera proxies, then Premiere Pro should automatically point you to the folder housing your full resolution media, which in most

cameras is also where the corresponding proxy clip will be.

For example, if you choose to attach a proxy to a RED clip, Premiere brings you to the clip's folder, and you can easily select the proxy .mov file right above the .R3D file and click OK.




If you're attaching proxies created in third party software, then use the Media Browser to navigate to their location, choose the matching clip, and click OK.

In either case, Premiere should be able to automatically locate the remaining proxy clips, assuming you've maintained the proper folder structure for in-camera clips or created a single proxy folder for third-party created clips. If Premiere doesn't Relink automatically, then you can repeat the process and link each clip manually.

When you've finished, your clips now have attached proxies that you can toggle between in both the Project Panel and the Timeline.

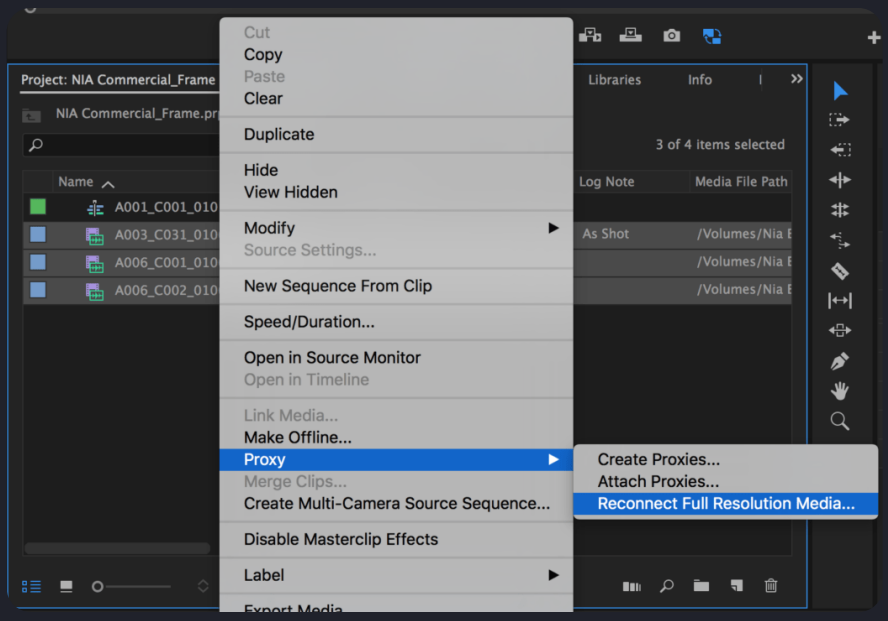
## 3. Reconnecting Full Resolution Media

The final workflow that Premiere Pro has accounted for is when you're given ONLY proxies initially and the high resolution in-camera media is delivered at a late date. This sometimes happens when media is being shuttled from on set to an offsite editor in an effort to save drive space and get

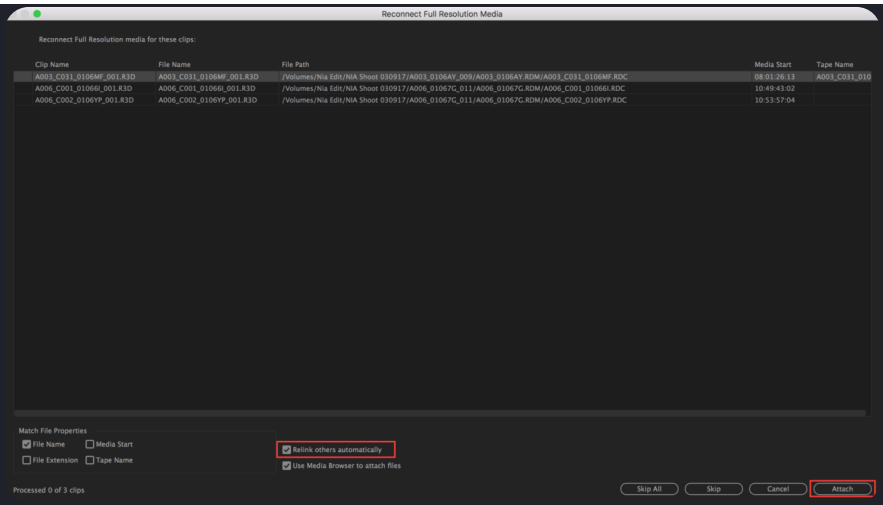
the editor working as quickly as possible.

In this scenario, we can begin our edit using the proxies we're given and then use Premiere Pro's Attach Full Resolution Media option to link our proxies back to their full res media when we receive it.

In the Project Panel, select the clips you want to attach full res media to and right click. In the context menu, choose Proxy –> Reconnect Full Resolution Media.




The Reconnect Full Resolution Media dialogue box opens, which works in the same way as the Attach Proxies dialogue box. Make sure the Relink others automatically box is checked and then, in the lower right corner, click the Attach button.



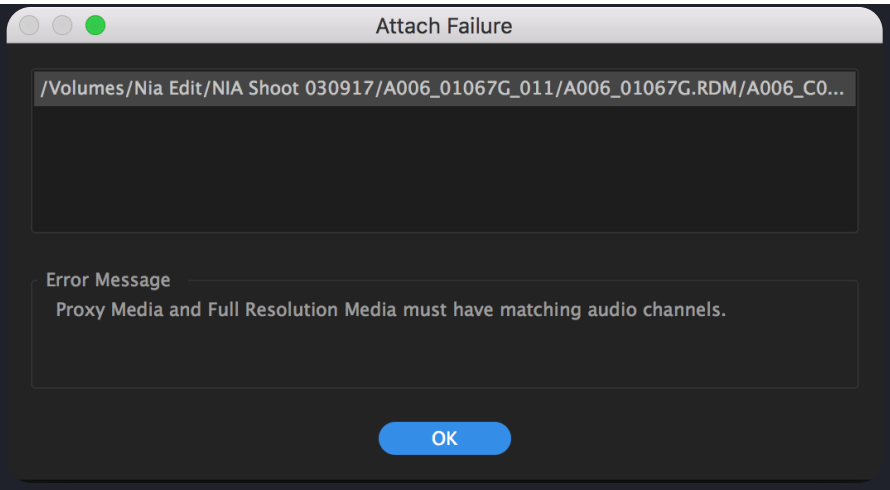

The Media Browser opens, allowing you to navigate to the corresponding full res clip. Once you've selected the correct clip, click OK.

Depending on the file structure of your camera media, Premiere Pro may or may not be able to automatically relink the rest of your clips. Automatically relinking isn't quite as reliable going from Proxy to Full Res media as it is vice versa.

If your other clips don't relink automatically, then repeat the process and manually attach each corresponding full res clip. It isn't fun, but trust me, it saves time in the long run.

## Mismatched Audio Channels Warning

When working with third-party software, it's imperative that the number of audio channels in your proxy media matches exactly with the number of audio channels in your full-res clips. If the two don't match, Premiere won't be able to attach the proxies. This limitation applies to both the Attach Proxies and Attach Full Resolution Media options.




I'm very much hoping that Adobe changes this limitation in future updates. That would open the door for both creating and syncing lower resolution proxies in third-party programs like Resolve or PluralEyes, while still being able to take advantage of Premiere Pro's proxy workflow.

If you're using any pro camera that has over 2 channels of audio (ie, C100, C300, FS5, FS7, Varicam, etc etc) even if you only recorded 2 channels of audio, you have 4 channels. Then, if you want to use a codec like h.264 for your proxies (for space requirements to send online, etc.) you CAN'T, because Premiere can only render h.264 in the MP4 container –and that container only supports 2 channels of audio. To add to the heartbreak of this workflow, when you try to create proxies with this workflow, Premiere doesn't even give you an error, it just doesn't start the proxy job.

This leaves us to use codecs like DNxHR LB or ProRes Proxy as our proxy codecs–which are great if you're actually using drives; but if you're trying to put a project on your laptop for vacation work, or sending online., it doesn't work as well.

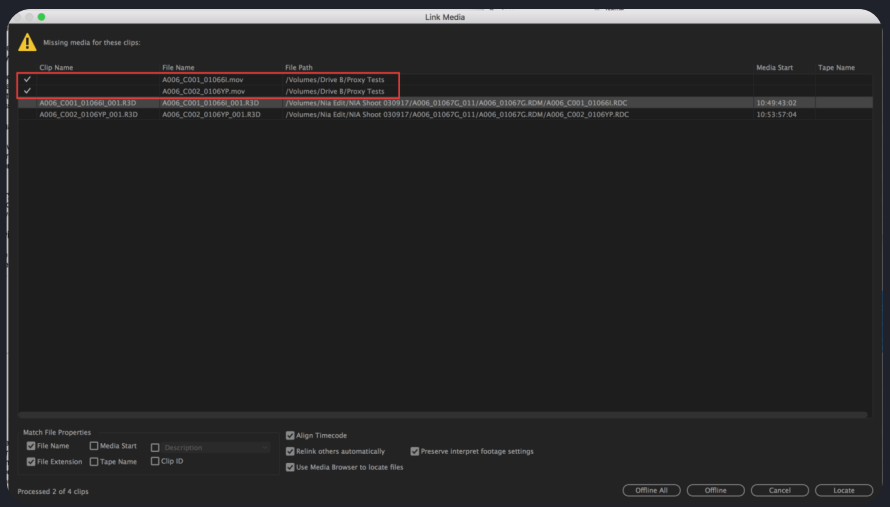
This is how I got around this limitation: you can use Davinci Resolve to generate proxies and use a Quicktime Filetype and then hh.264 codec inside that filetype. Then, you'll have to attach proxies manually to the full-res media in Premiere, but it works.

# Copying Proxy Files And Relinking

As we've mentioned, one of the most useful features of this workflow is that we can take our lightweight proxies on the go to either edit from lower-powered laptops or to quickly share media and projects between team members.

Copy the folder containing your proxy media over to a portable drive. Don't forget to copy your project file over as well.

With only the portable drive connected, launch the project. You'll see the familiar Link Media dialogue box. Premiere has gotten pretty good at locating media moved across drives, so chances are, it'll locate your proxy clips and relink them as shown below, giving you the option to Locate, Cancel, or Offline the full res media.




- Cancel allows you to reconnect back to your high res media at a later date. Each time you open the project file, you'll get the option to relink.
- Offline or Offline All tells Premiere Pro to mark the high res

media as offline and the Link Media dialogue won't open again. You can still reconnect your high res media manually at a later date.

- Locate doesn't apply in this situation because we don't have the full res media with us.

## Tips for Working with Proxies in Premiere Pro

- When you right click and get Properties on a clip with a proxy attached, you'll first see information for the full res clip and below that, information for the proxy clip.

Properties: A001_C001_01052V_001.R3D

File Path: /Volumes/Nia Edit/NIA Shoot 030917/A001_0105UX_001/A001_0105UX.RDM/A001_C001_01052V.
Type: RED R3D Raw File
File Size: 2.51 GB
Image Size: 5120 x 2700
Frame Rate: 23.976
Total Duration: 00:00:23:08
Pixel Aspect Ratio: 1.0

Red R3D Details:

Camera Firmware Version: 6.3.75
Camera Sensor: DRAGON 5K S35
Minimum Color Version: 2
Resolution: 5120x2700
FrameRate Num: 24000
FrameRate Den: 1001
Reel ID: A001_C001_01052V
Timecode Preference: Absolute
Edge Timecode: 01:00:00:00
Absolute Timecode: 20:04:29:14
Camera ID: A
Clip ID: 001
Brightness: 0
Contrast: 0
Digital Gain Blue: 1
Digital Gain Green: 1
Digital Gain Red: 1
Exposure Compensation: 0
Exposure Time: 2.085e+04
GenLock Setting: 0
GMT Date: 20000105
GMT Time: 200541
ISO: 1600
Jam Sync Setting: 0
Local Date: 20000105
Local Time: 200541
Saturation: 1
Shutter Degrees: 180.2
Shutter Fractions: 47.95
White Balance Kelvin: 6000
White Balance Tint: 0

Proxy Media

File Path: /Volumes/Nia Edit/Proxy Test/A001_C001_01052V_001_Proxy.mov
Type: QuickTime Movie
File Size: 46.03 MB
Image Size: 1024 x 540
Frame Rate: 23.976
Source Audio Format: 48000 Hz – 16 bit – Stereo
Project Audio Format: 48000 Hz – 32 bit floating point – Stereo
Total Duration: 00:00:23:08
Pixel Aspect Ratio: 1.0

QuickTime Details:
Movie contains 1 video track(s), 1 audio track(s), 0 closed caption track(s), and 1 timecode track(s).

Video:
There are 560 frames with a duration of 1001/24000ths.

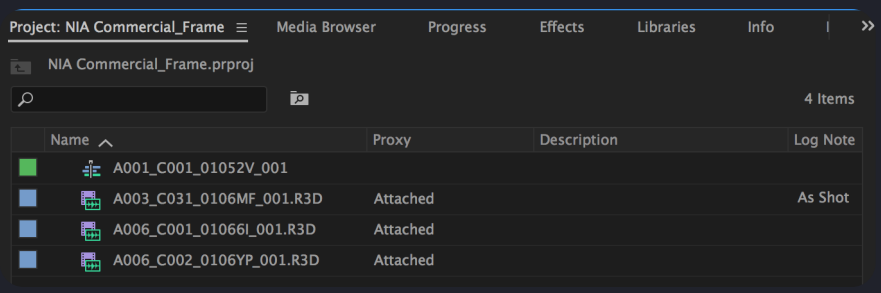
Video track 1:
Duration is 0:00:23:08
Average frame rate is 23.98 fps

Video track 1 contains 1 type(s) of video data:

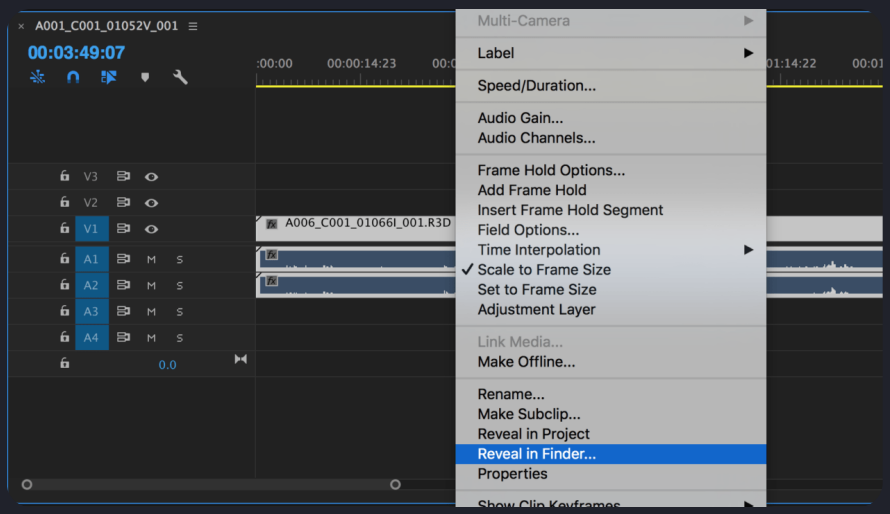
Video data block #1:
Frame Size = 1024 x 540
Compressor = Apple ProRes 422 (Proxy)
Quality = Most (5.00)

- Premiere always uses the full res clips when rendering previews, exporting media or using video effects that require Premiere to analyze a clip.
- You can add a Proxy column to the Metadata Display in your project panel. This column can read either Attached or Offline, indicating the status of your proxies, however this feature can be finicky. It's often blank, even when proxies are indeed attached. Alternatively, use the method below to

double check that proxies are working.



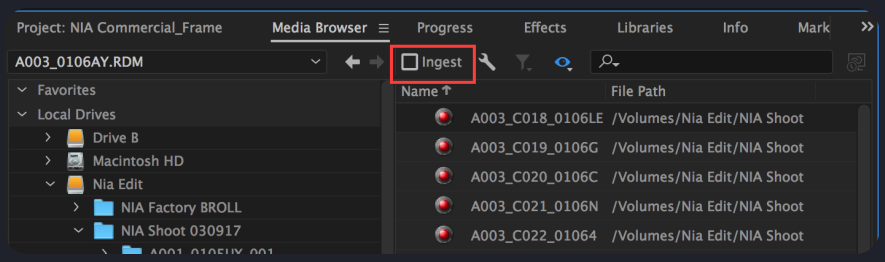

- If you can't tell the difference by eye between your full res and proxy clips and want to make sure they're both attached, you can use the Reveal in Finder command as a dummy check. Toggle your proxies on and then right click a clip either in the project panel or the timeline. In the context menu, choose Reveal in Finder. If the clip that's revealed is your proxy clip, then you know it's attached correctly. You can perform the same test on your full res clips by toggling them on.




- When you create proxies in Premiere Pro either on Ingest or via the Create Proxies function, your files are appended with "_Proxy" and placed in a folder named "Proxies." While Premiere Pro recommends this labeling system for best relinking results, it's not required. If you don't use this naming convention, we do recommend making sure your proxy file names match exactly to your in-camera clips, with the exception of their file extension. For example, A006_C001_01066I_001.R3D and

A006_C001_01066I_001.mov.

- Remember to uncheck the Ingest box in your Media Browser Panel, or under File –> Project Settings –> Ingest Settings, when you want to import media without creating proxies. Note that when you create a new project, the Ingest option will default to unchecked, even if your last project had it checked.




## Wrap Up

Now that you're armed with a complete understanding of Creating Proxies, Attaching Proxies, and Reconnecting Full Resolution Media in Premiere Pro, you can dive in at nearly any stage in the process to supercharge your edit speed, skip the online entirely, and take your project with you on that beach vacation.

## Avid Media Composer Proxy Workflows

With its built-in proxy workflow, Avid Media Composer makes editing with high-resolution media as painless as possible – you can do it all without ever leaving the program or interrupting your work.

By the end of this article, you'll understand how to use the source browser to link to high-resolution camera media,

transcode that media to lower resolution proxies, and relink back to the original media when necessary.

We'll also talk about Avid's proxy mode timeline setting, which we can combine with proxy media to get even better performance.

The first step to get your proxy groove going is to bring in high-resolution media files to your Avid project.

## Importing vs Linking

There are two ways to bring media into an Avid project:

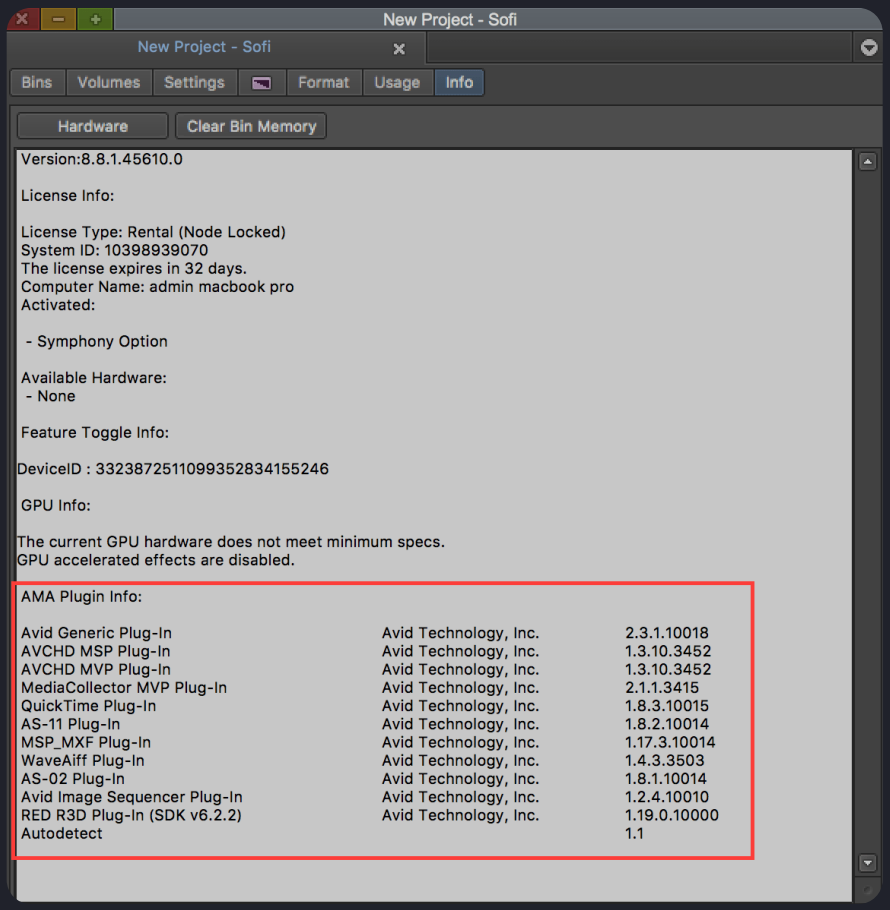
- Link Media. Linking to source media allows us to take advantage of AMA (Avid Media Access), which links the file-based media directly into a bin through a plug-in, leaving it in place on the hard drive and maintaining all file and folder structures.
- Import Media. Importing source media creates new MXF files that are placed and managed within the Avid MediaFiles folder, on whichever drive you've specified in your Import Settings, and does not maintain original file and folder structures.

When working with high-resolution camera media, it's generally best practice to maintain the original folder structure, especially if you'll be color correcting in another application like DaVinci Resolve. In some cases, linking also provides more control over your media, like with RED footage, where linking with AMA allows you to access and edit color profiles for the raw media.

# Installing AMA Plugins

Before we can actually link media, we need to download and install the correct AMA plugin for our source media. Though a few standard plugins come bundled with Avid, like one for Quicktime Media sources, most have to be installed manually, which allows camera manufacturers to update them with each new camera release.

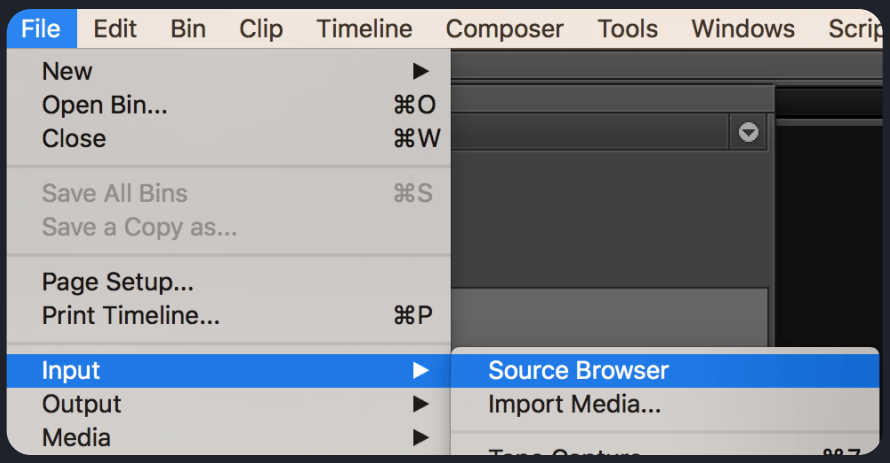
You can find a list of downloadable plugins on Avid's AMA page. If you're not sure if you've already installed a specific plugin, you can check by opening the Project Window (⌘9) and clicking the Info Tab.



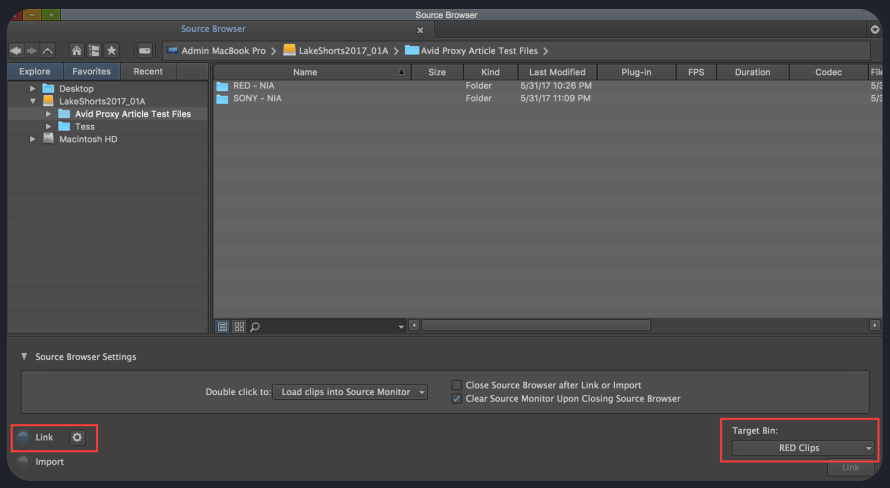

Once you install the plugin for your specific camera media, the process of linking and creating proxies is the same for each plugin.

# Linking File-Based Media with AMA

Open the Source Browser (File > Input > Source Browser), which allows us to browse for files to either import to or link to.

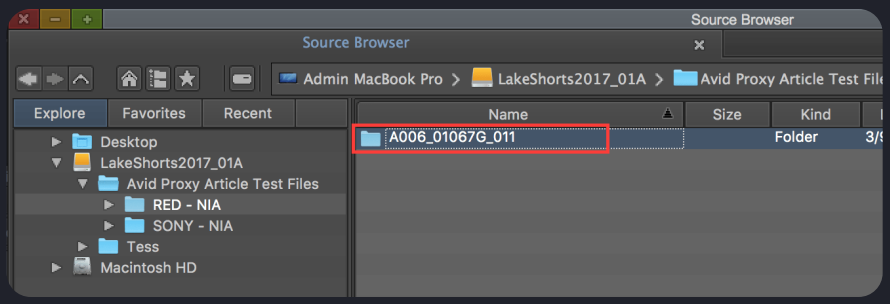



Make sure the Link option is selected in the lower left corner. In the lower right corner, notice the setting for Target Bin. In the drop-down menu, you can set which bin you'd like your linked clips to end up in or even create a new bin.

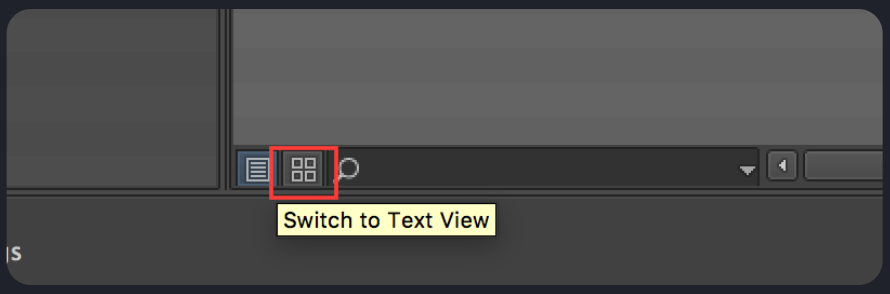


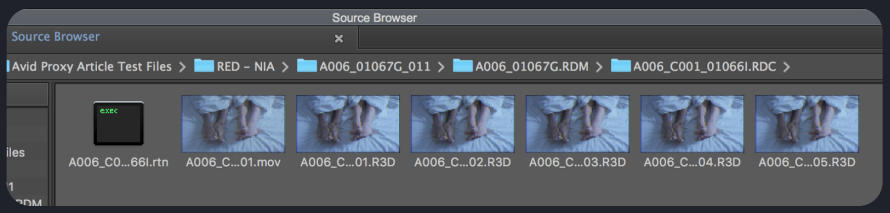

Once you've set your target bin, use the left-hand side of the browser to navigate to your source media. In this example, I'm using RED media, which like most high-resolution cameras, uses a very specific file structure that includes many folders. In most situations, you can select the outermost folder(s) and all

the clips inside will be linked.

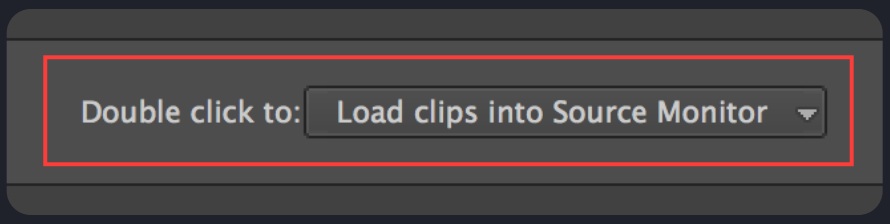



If you prefer, you can also select the thumbnail view option and preview clips.



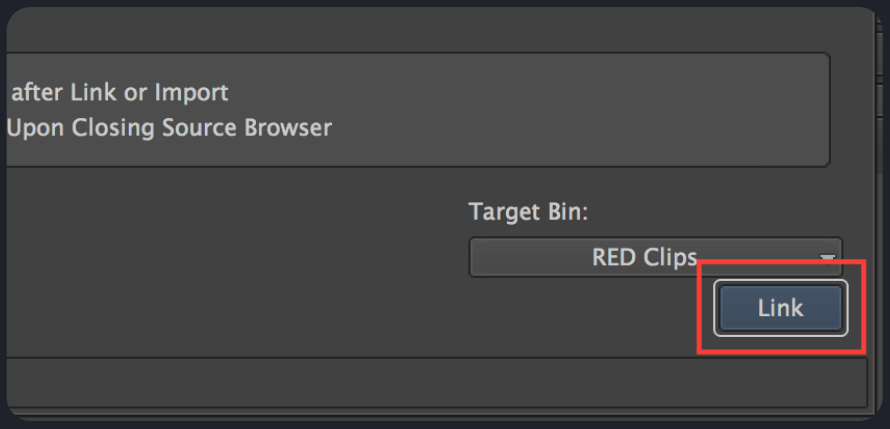

For an even better view, set the drop-down menu next to "Double click to" to "Load clips into source monitor."



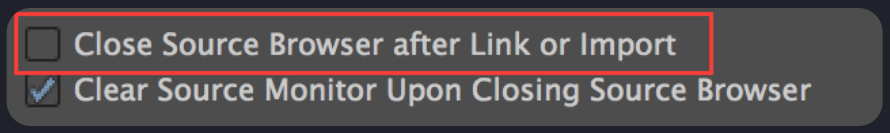

Now you can playback clips in the source monitor by double clicking them, without actually bringing them into the project.

Once you've selected your clips, click the Link button in the

lower right corner and they'll populate the specified bin.

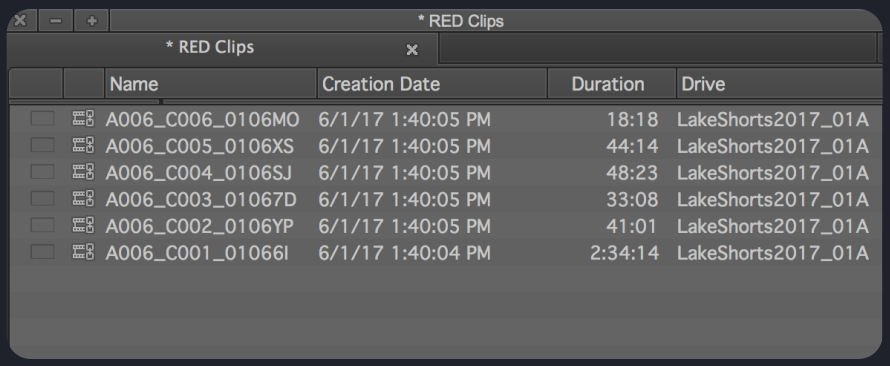



Note: If you want the Source Browser to close automatically, select the Close Source Browser after Link or Import option.




## Transcoding AMA-Linked Media in the Bin

Your high resolution linked media will now be available in the bin you specified.

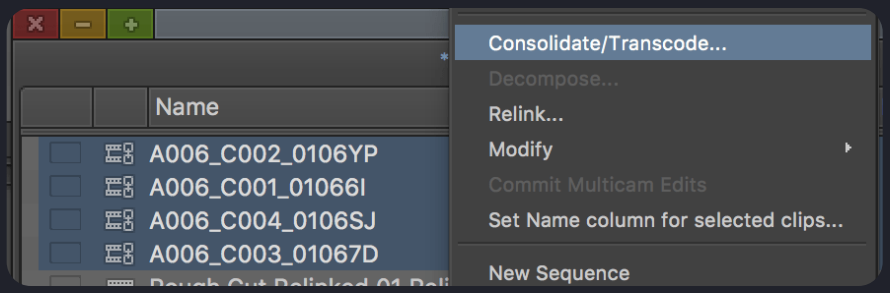



Now we're ready to transcode the source media to a more portable, edit-friendly, proxy resolution.

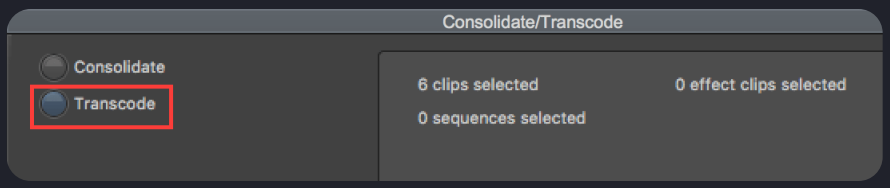
While I like to transcode right after linking, you can actually begin to edit the source clips and transcode at any time if you prefer. For example, you can string out a sequence of selects and then transcode just the clips in that sequence to save time and drive space.

I don't recommend getting too deep into organizing your clips before you transcode. Transcoding in Avid works bin by bin, so if you organize a project into scenes before you transcode, you'll have to select the clips in each bin separately (or make a sequence containing the clips you want) to transcode, which can get tedious.

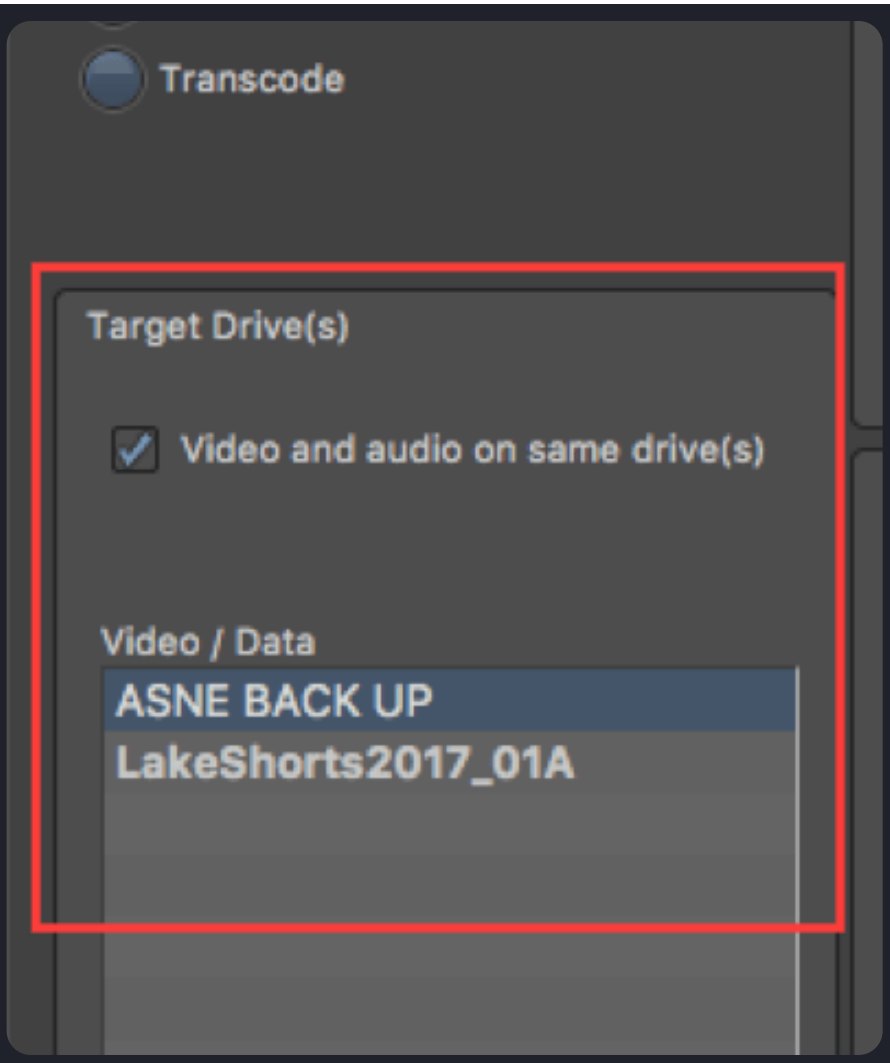
In the bin, select all the clips (or a sequence) you want to create proxies for, right-click and select Consolidate/Transcode from the context menu.




In the dialogue box that opens, select the Transcode button in the upper left-hand corner.




Below the Transcode button, you'll see an option to select Target Drives. Avid treats transcoded proxies like imported media, meaning it places and manages them within the Avid MediaFiles folder, so we have to specify where we'd like that folder to go.



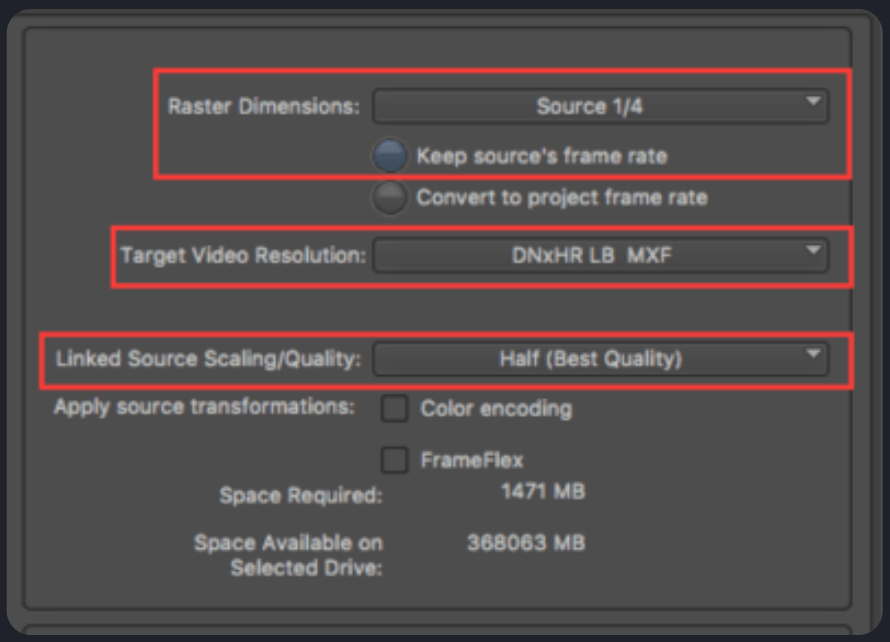

I prefer to transcode directly to a portable drive so that my source media and proxy media are in separate places. That way, I can easily take my portable drive with me. But if you don't have a separate drive available, or are transcoding for performance rather than portability, you can transcode to the same drive as your source media. You can always copy your proxy media to a different drive later if you change your mind.

Once you've selected your target drive, choose an option under the Raster Dimensions drop down menu. This will determine the frame size of your proxy media. 1/4 is usually a good choice (and allows us to best take advantage of the timeline's proxy mode, explained later on), though it will depend on what frame size you're starting with. In this example, I'm working with 4K RED files.

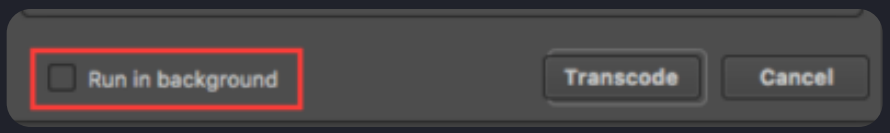
Under Target Video Resolution, select DNxHR LB, which is a low bandwidth, lightweight codec.

In most cases, you'll want to select the "Keep source's frame rate" option so that you don't have any issues relinking back to the high-resolution media.

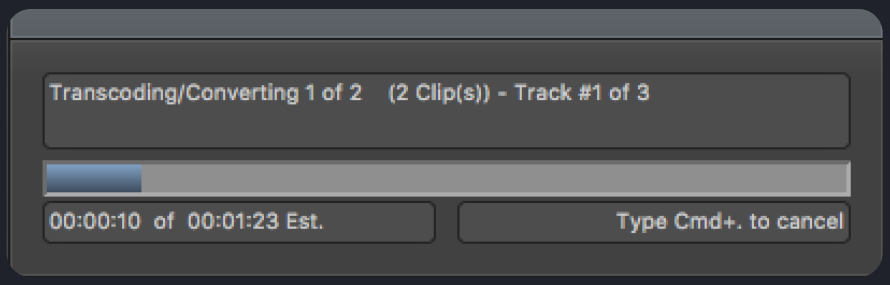
Under the Linked Source Scaling/Quality drop down menu, I generally choose Half (Best Quality), but you can choose whatever you'd like. This setting affects the playback quality of the proxy media, so depending on your computer's RAM and graphics card, choosing Full may impact how smoothly your footage plays. If you're starting with 6K media, then I probably wouldn't go above Half, whereas if you're starting with 2K, you'll likely be able to play back well at Full.



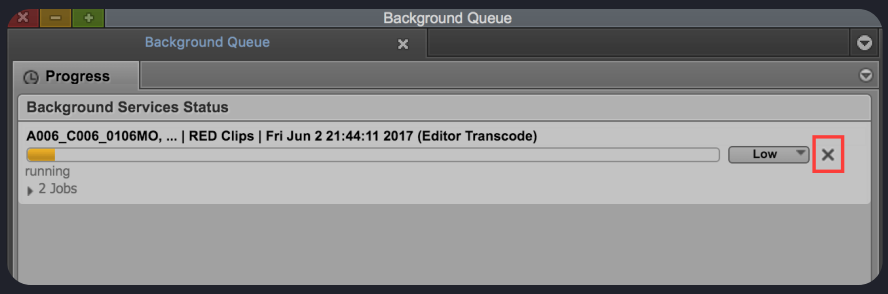

Finally, at the bottom of the Consolidate/Transcode dialogue box, you'll see a checkbox for "Run in background."



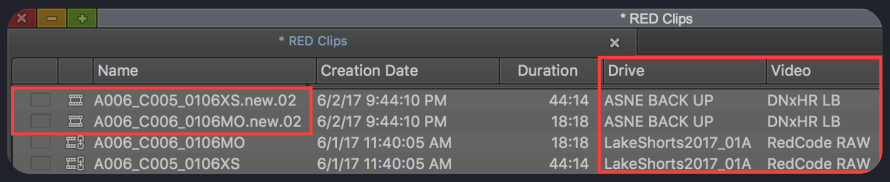
If you leave this option unchecked and then click Transcode, a progress bar will come up while Avid creates the files and you'll have to wait for the transcode to complete for continuing to work in the project.




If the box is checked, Avid will transcode your files in the background and you can keep working in your project while you're waiting. I like this options because you can check the progress of each clip being transcoded by opening the Background Queue under Tools > Background Queue. You can also cancel the transcode here by clicking the X on the right-hand side of the progress bar.




Either way, once your transcodes finish, you'll notice that the proxy clips appear in the bin with a .new extension. In the Drive column, you can verify that the .new clips are on the drive you chose for your proxies and in the Video column, you can see that the clips are DNxHR LB.



To avoid confusion, move the proxy clips into a new bin that's clearly labeled.

Now you can organize and edit your project with the proxy media for improved speed, or take your drive with the proxies on the go, leaving the original high-resolution media behind.

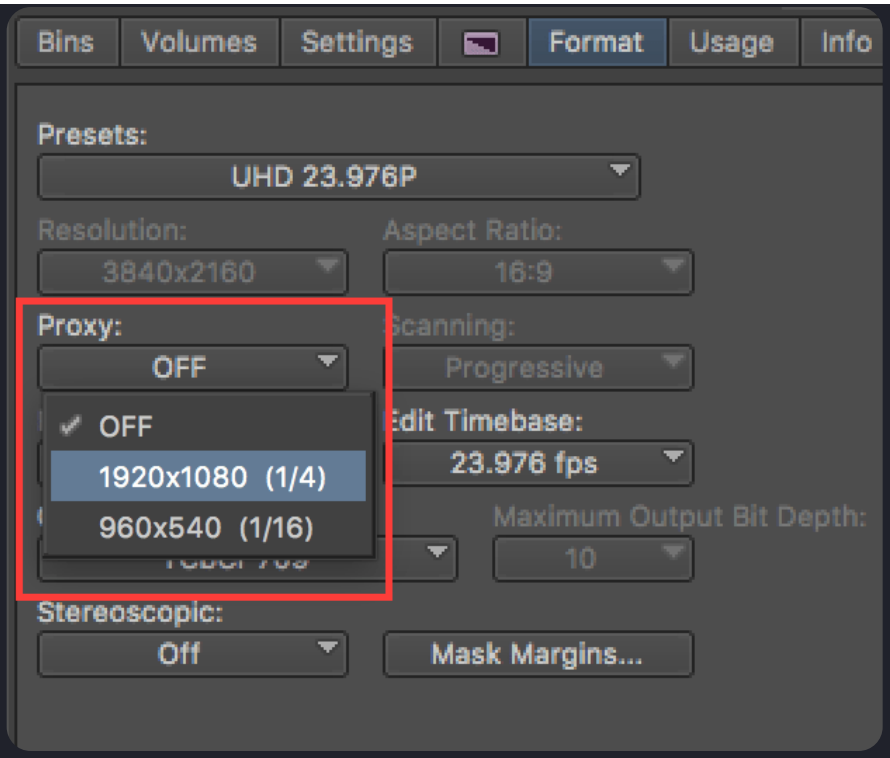
## Setting a Proxy Mode for the Timeline

In addition to working with transcoded proxy media, you can also switch to a proxy mode in the timeline to playback high-resolution media at a lower resolution.

This proxy mode is only available for high-resolution projects (not for HD projects) and is different than the video quality options (yellow/green modes) because your sequence will render at whatever proxy quality you've selected.

To get the best performance, you can first transcode your media to a proxy resolution (like 1/4) as described above and then set your timeline to playback at the matching proxy resolution (1/4). That way, there's no real-time processing when the clips are played.

To set the timeline's proxy mode, open the Project window (⌘9) and select the Format tab. Under the Proxy dropdown, you can choose to play back your clips at 1/4 or 1/16 of their original resolution.




Any clips that are resized will have a spatial adapter applied, which is shown by a green dot on the clip in the timeline.

When you render or export your sequence, it will use the proxy resolution that you've chosen. You can change the proxy mode anytime, but any previously rendered media will go offline unless you switch back to that proxy mode.

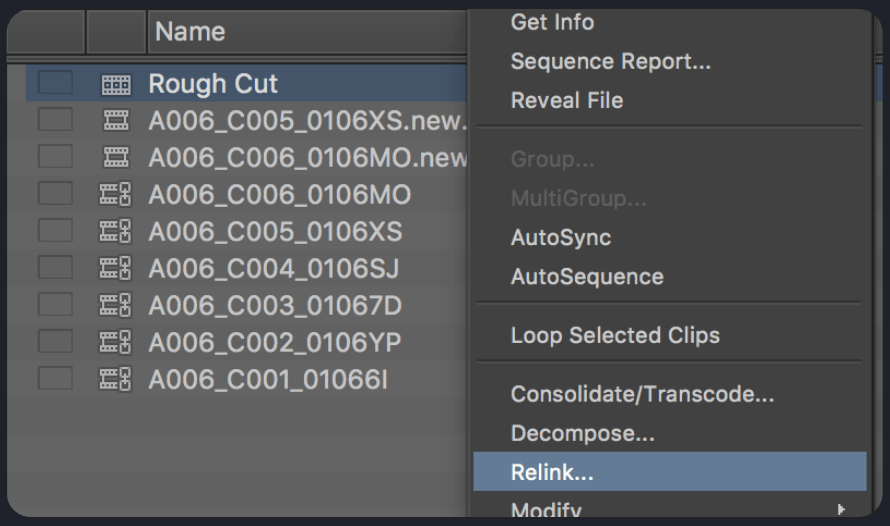
Remember that when you want to export your sequence, you need to turn off the proxy mode to set your media back to its full resolution.

## Relinking to High-Resolution Source Media

Once you're done editing your project with proxy media, you'll want to switch back to the high-resolution media before outputting your final sequence.

Right click on your final sequence and select Relink from the

context menu.

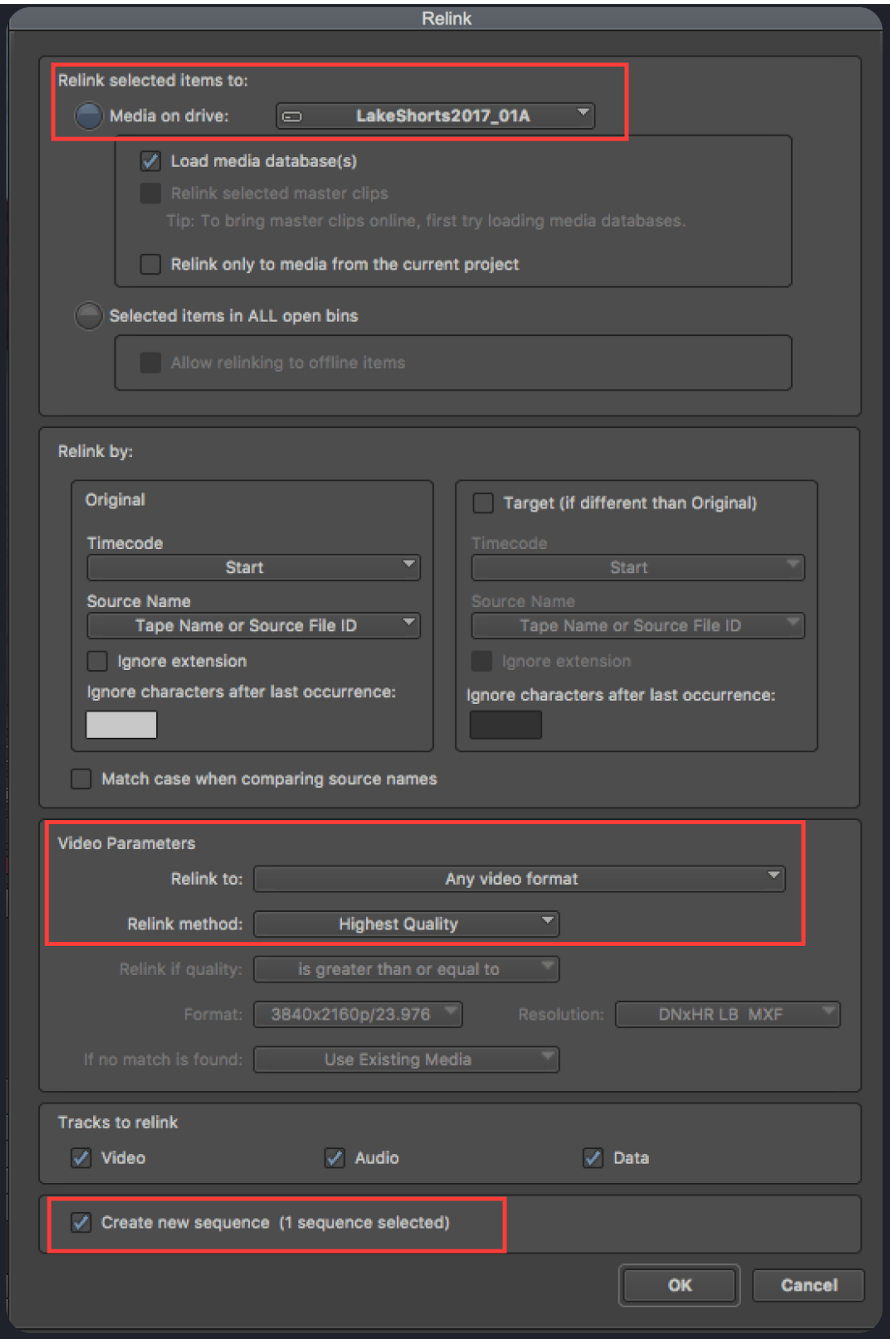



In the Relink dialogue box, you can choose which drive your high-resolution media is located on under the "Media on drive" drop-down menu, or leave it set to All Available Drives.

Under Video Parameters, set:

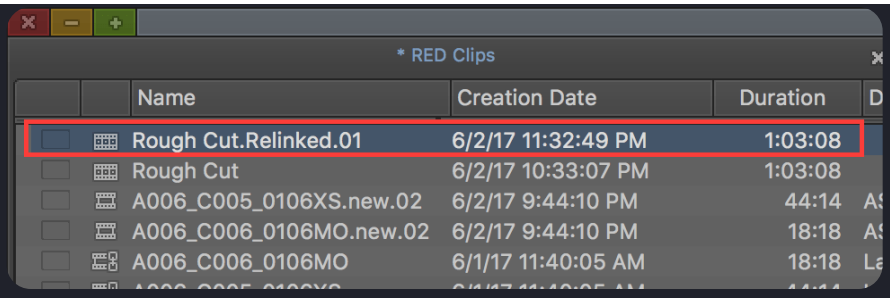
- Relink to: Any video format
- Relink method: Highest Quality.

At the very bottom of the dialogue box, check the "Create new sequence" box.




Click OK.

A new sequence with a .Relinked extension will appear in the bin, using the original, high-resolution AMA-linked clips.




You can use this new sequence to do all of your final outputs at the highest quality.

## Relinking to Proxy Media

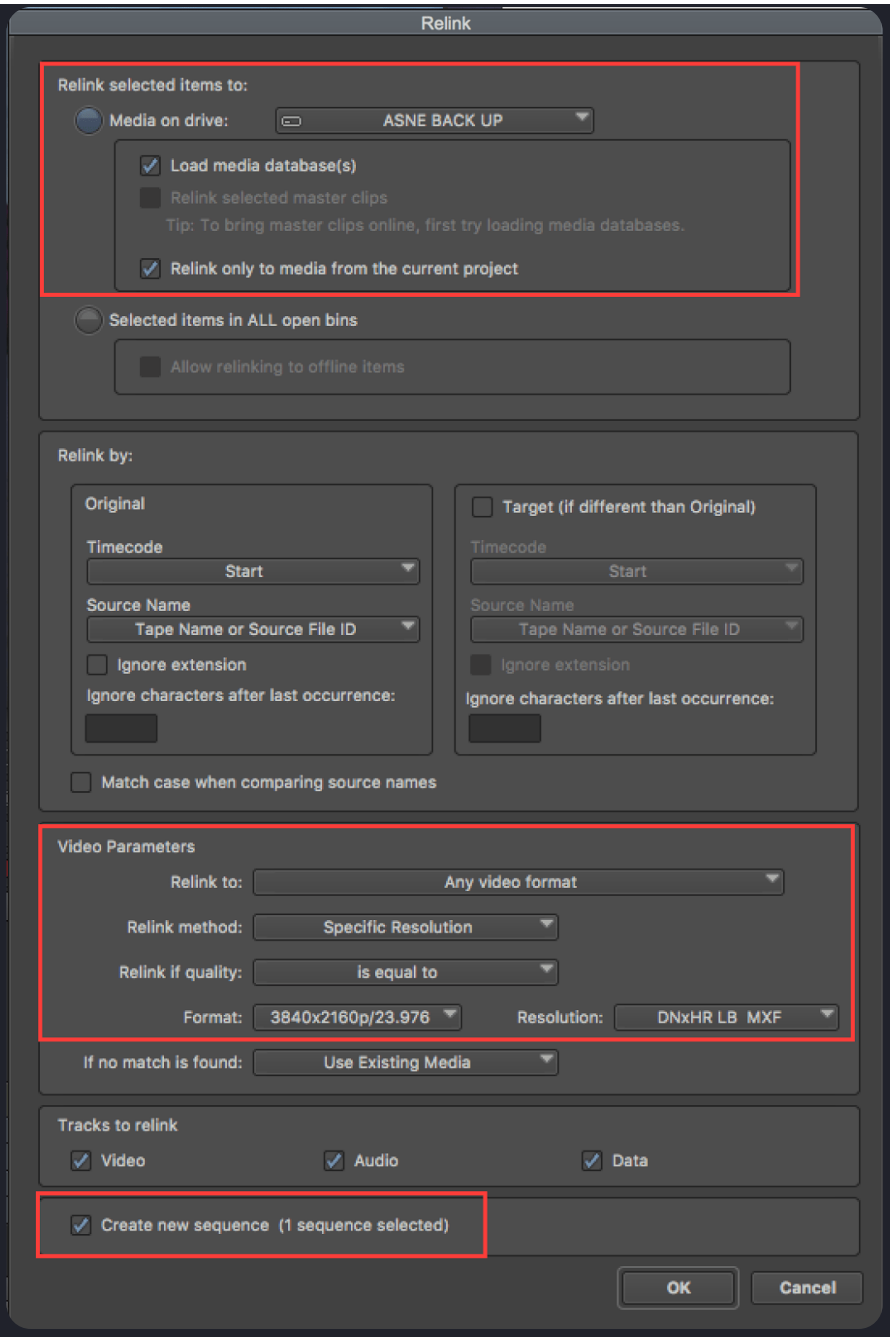
You can also relink a high-resolution sequence back to existing proxy media, if necessary.

As described above, right click on the sequence in the bin and select "Relink from the context menu." In the Relink dialogue box, choose which drive your proxy media is located on under the "Media on drive" drop-down menu, or leave it set to All Available Drives. Check the boxes for "Load Media Databases" and "Relink only to media from the current project."

Under Video Parameters, set:

- Relink to: Any video format
- Relink method: Specific Resolution\
- Relink if quality: is equal to and choose the correct Format and Resolution

At the very bottom of the dialogue box, check the "Create new sequence" box.




Click OK.

A new sequence with a relinked extension will appear in the bin, now using the proxy media.

MultiRez and Dynamic Relink allow you to create two sets of media files—for example, a high res and low version—that apply to the same master clips. You can edit with the low-res media and just use the high res media for output, and you can automatically switch from one to the other without having to use the Relink command.

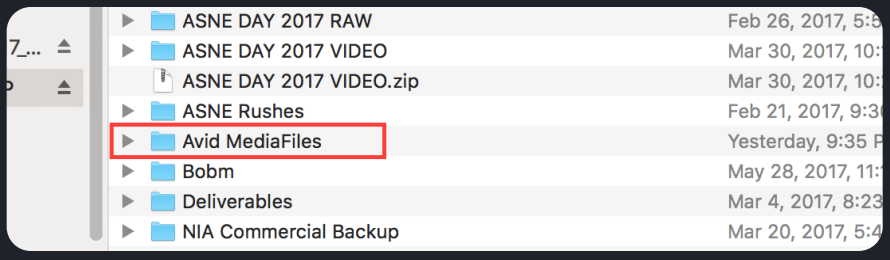
To take advantage of MultiRez and Dynamic Relink, you have to have the Avid Interplay Media Indexer installed.

Interplay, Avid's MAM (Media Asset Manager), is a high-end product that comes with dedicated support from Avid, and so we assume that, if you have an Interplay system, you probably don't need this guide. For more about this workflow, see Avid's help documents here and here.

## Taking Your Proxy Media on the Go

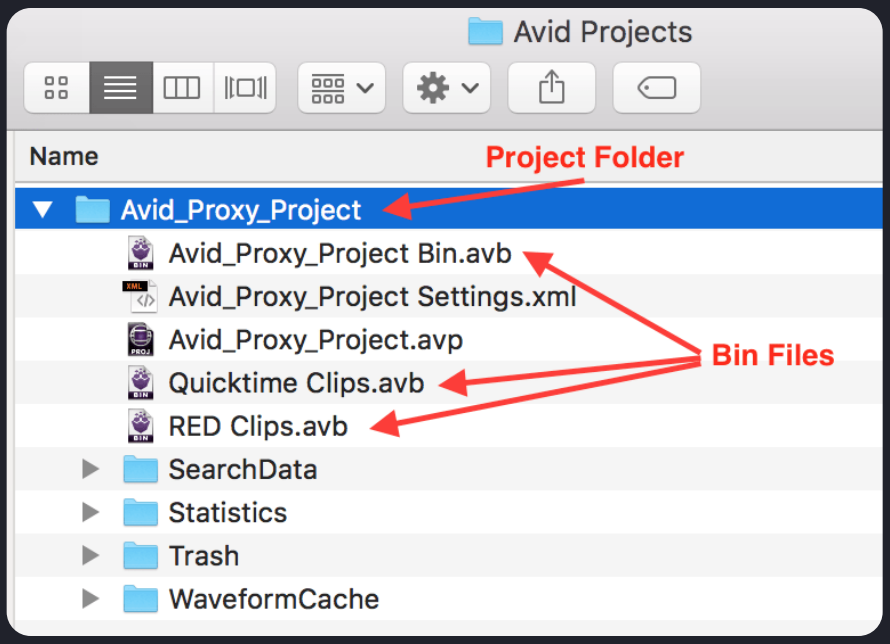
Avid is particularly well-suited to sharing lightweight proxy files with others or moving them between small, portable hard drives because all you need to get a project up and running is the bin (.avb) file with your clips and sequence.

Sharing proxy media between drives is as simple as copying over the root-level Avid MediaFiles folder on the drive with your proxies.




You can copy an entire project by selecting the project's outermost folder (the name of the project) to a new drive, or

you can choose to copy over a specific bin or bins that you can then open in a new project.




## Wrapping Up

Avid's proxy workflow is both easy and integrated; you can go from start to finish without ever leaving the program or holding up your work.

Combined with Avid's simple project and bin structure, taking your work on the go is as easy as copying over the project or bin you want and taking your proxy media drive with you.

These three guides contributed by Sofi Marshall, Freelance Editor.

## Davinci Resolve Proxy Workflows

It's no secret that Resolve requires some powerful hardware. Whereas other NLEs can comfortably run on laptops, Resolve

has a legacy of big iron deployments. Before da Vinci Systems went bankrupt and had its assets sold off to Blackmagic Design, Resolve only existed as a turnkey solution, with vendors supplying specific hardware and software. Their hardware was quite powerful for its time, and Resolve ran on specially tweaked distributions of Linux.

Blackmagic Design released DaVinci Resolve to the masses by porting the software to Mac and Windows, and allowed users to install it onto their own hardware. This was hugely liberating –whereas color correction had been a kind of secretive dark art hidden away in color suites costing hundreds of thousands of dollars, the hardware and software associated with legitimate color correction dropped by an order of magnitude. Soon, a proper color correction suite could realistically be built for less than $50,000.

Much has been written about how Resolve is perhaps turning into the one single most powerful tool in post-production. After Fairlight was incorporated into 14, Patrick Inhofer wrote about how Resolve actually has a shot at turning into the long elusive "Studio" software: one integrated app wherein different post-production specialties would all be able to collaborate within the same, integrated software: dailies, offline editing, color grading, audio mixing, and online editing. The release of 15 brought the Fusion page in, and with it, VFX artists.

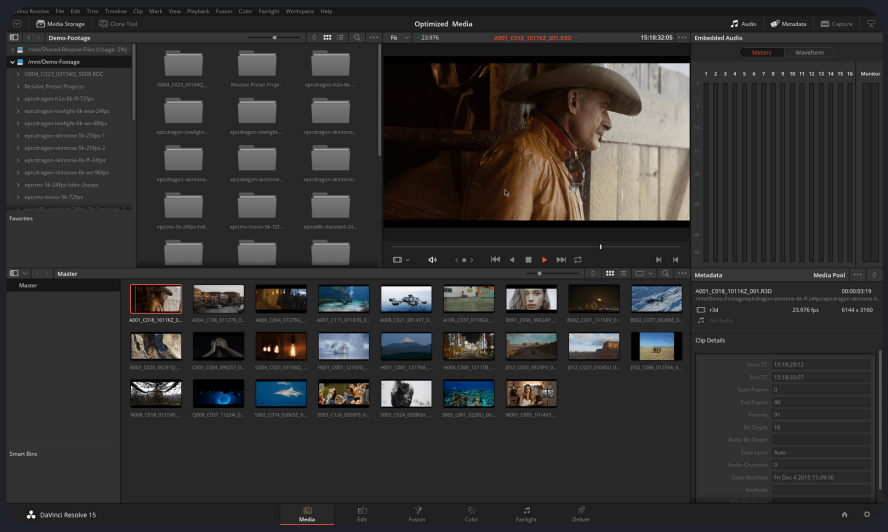
So Resolve is poised to be used far and wide in assorted post-production specialties–but one big hurdle is to make sure that it's properly tuned for every different kind of collaborator.

Offline editors and audio mixers don't want or need the hardware to be natively debayering 8K 60-fps RED WEAPON footage. It would be a ridiculous waste of resources to buy a high-end finishing workstation hardware just to allow every collaborator to edit from original camera files.

Instead, just as Avid Media Composer, Adobe Premiere Pro, and Apple Final Cut Pro X all have ways of internally generating proxy files, so too does DaVinci Resolve.

In DaVinci Resolve, both in the interface as well as the documentation, these kinds of internal proxies are referred to as "optimized media."

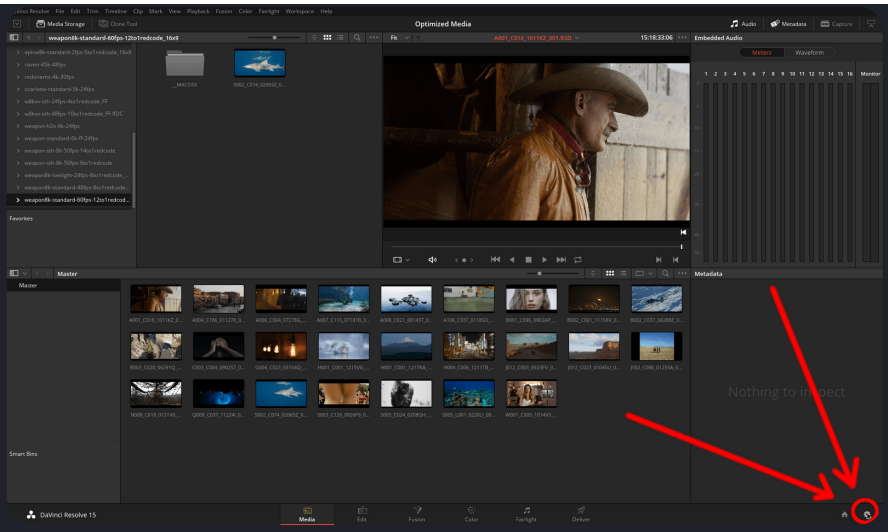
Let's take a look at some heavy, sample R3D files. It can be quite taxing to debayer and play such files in their native format.



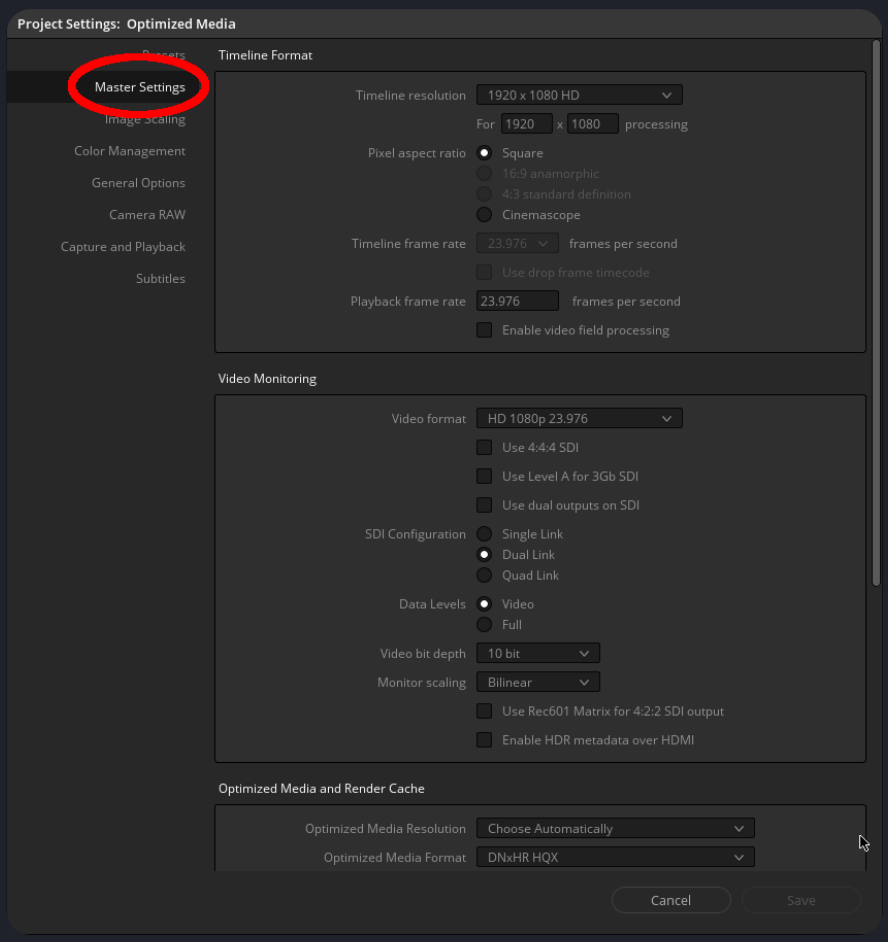

Here in the media page we can perform operations on specific clips that follow those clips around to the other pages, in the Edit, Color, Fusion, Fairlight, and Deliver pages.

Before we go creating optimized media though, we need to make sure we're putting it into the correct spot.

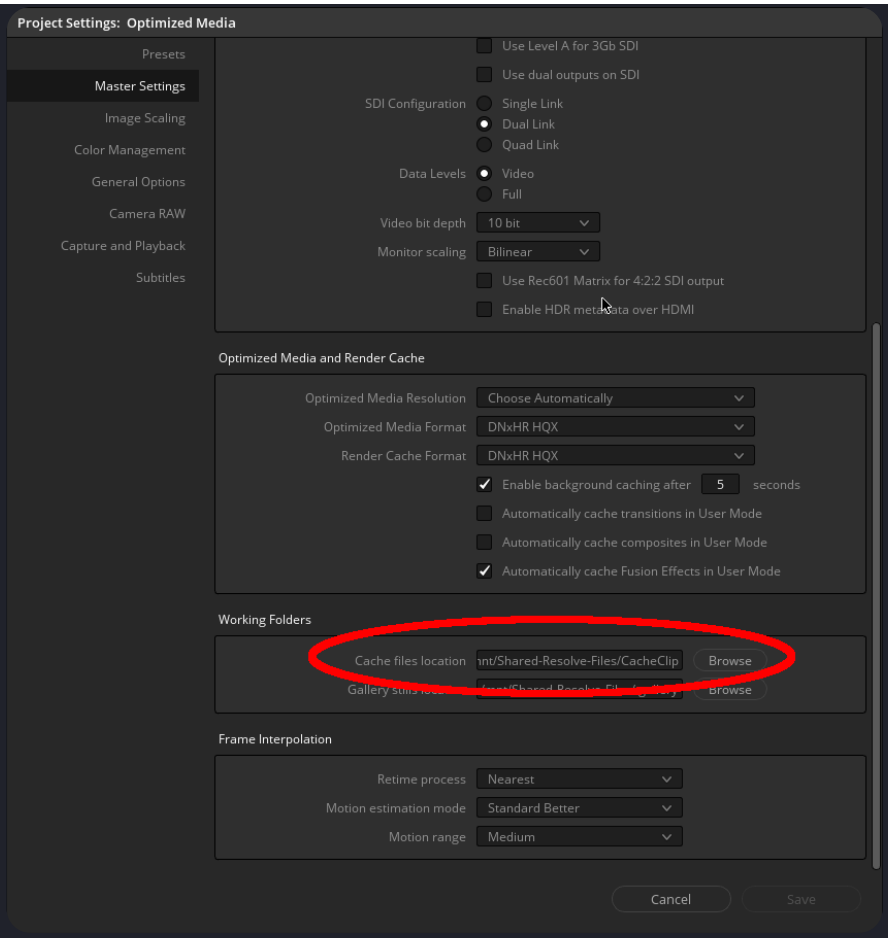
Go into "Project Settings," either by clicking File > Project Settings or by clicking the sprocket icon in the bottom right of the GUI.




Once in Project Settings, you'll want to be in the "Master Settings" section on the left.

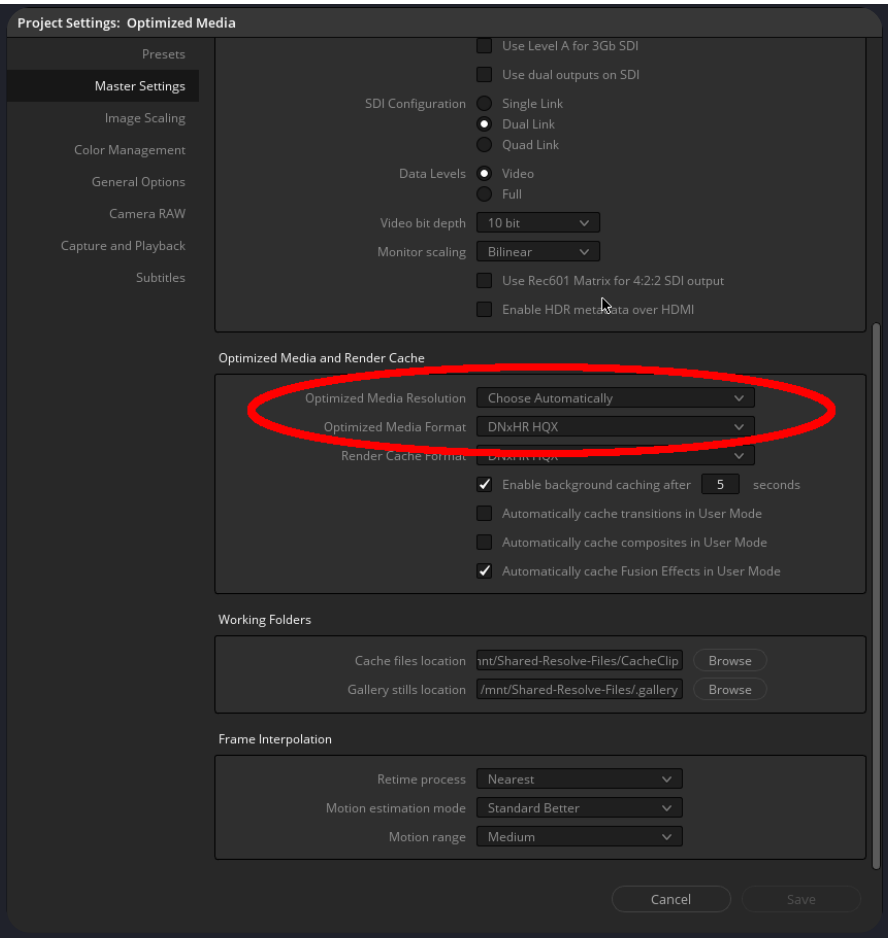



In this window, scroll down to the "Working Folders" section.




The "Cache files location" is where the optimized media will live. We'll want to put it on either a proper network share from a NAS or SAN, a connected direct-attached external hard drive or RAID, or even an internal dedicated hard drive or RAID, separate from the boot drive. Wherever you choose, you'll need good read and write speeds.

Hop back up to the previous section, "Optimized Media and Render Cache." We'll need to tweak two settings here, for "Optimized Media Resolution," and "Optimized Media Format."




In "Optimized Media Resolution," you can manually choose what size you want these proxies to be. So if you have UHD (3840 x 2160) footage but want your proxies to be 1080p (1920 x 1080), you'd choose "Half." The fractions refer to the relative sizes of one dimension like height or width, but "Half" actually provides a quarter of the number of pixels. We're in two dimensions, so $(½)^2$ is equal to ¼ the number of pixels.

"Choose Automatically" is another option. The DaVinci Resolve 15 manual explains:

The "Choose automatically" option of the Resolution setting bears a bit more explanation. When selected, only source media with a higher resolution than the selected Timeline resolution will generate downsized optimized media. How much each clip will be downsized depends on how much larger each clip is than the Timeline resolution. For example, if you're working within a 1080 resolution project, then 8K clips

will generate quarter-resolution optimized media, and 4K clips will generate half-resolution optimized media, such that all optimized media is somewhere around 1080 resolution. All clips that are 1080 and smaller generate optimized media at the same resolution as the source clips.

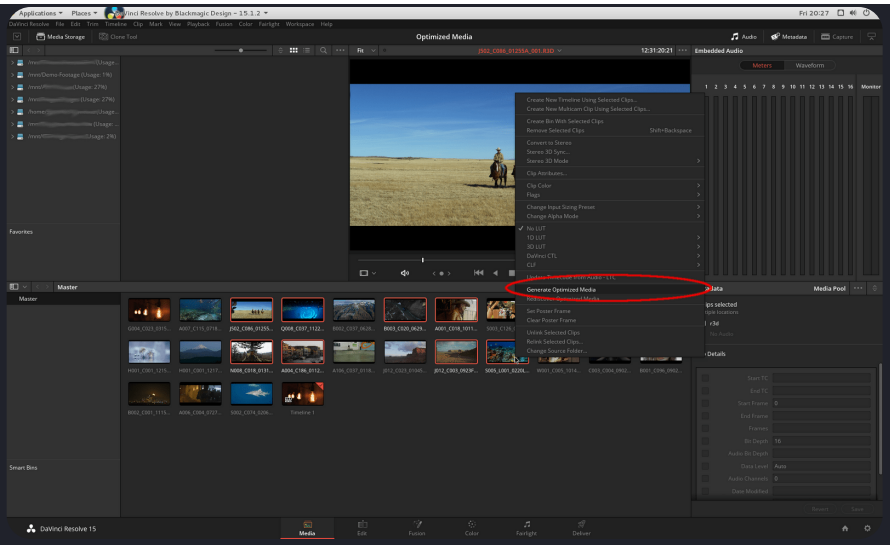
You have lots of options to choose a format in "Optimized Media Format." If you're just cutting the offline edit or mixing audio in the Fairlight page, you might choose DNxHR LB or ProRes 422 Proxy. However, if you intend to use this optimized media in your final color on the Color page or VFX in the Fusion page, you'll want to choose a high-quality mezzanine format.

Now that we have our settings in place, we can close the Project Settings window and return to the Media Pool.

For which clips do we want to generate optimized media? In the Media Pool, you can either:

- Click on one clip;
- Click on one clip, hold down shift, and click on another clip to select all clips in between, including both clips you selected; or
- Hold down Ctrl [on Windows or Linux] or ⌘ [on Mac] and click on particular clips you want to select.

When you have one or more clips selected, right-click and select "Generate Optimized Media."




Then you'll see a window showing you a progress bar.

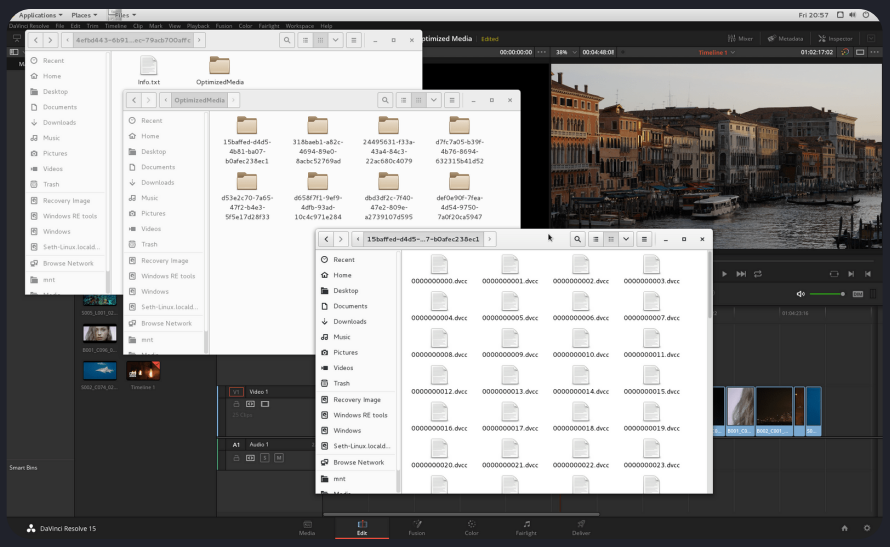
Unfortunately, in Resolve, this can't be a background process, like in Final Cut Pro X. So go grab a coffee.

When you return from coffee, you'll find that, as you add clips into the Edit page, you'll be using the optimized media. The associated optimized media will be associated with the clip inside Resolve.

Over on the Edit page, select "Playback" in the menu bar and make sure that "Use Optimized Media If Available" is checked. If a particular clip in your sequence has associated optimized

media, then that's what you'll be playing back.

If you went looking for the optimized media in File Explorer or Finder, you'd find that the optimized media exists in the folder you selected earlier, but it's not particularly useful outside of DaVinci Resolve. It exists in a proprietary .dvcc image sequence container.




Contributed by Seth Goldin, Tech, post-production, and workflow at Freethink Media.

Later on (usually during the conform), the proxies will be discarded and the project will be linked back to the original camera files before being sent to color correction or final delivery. Once the project is reconnected to the original camera files, it is back "online."

## Why edit offline?

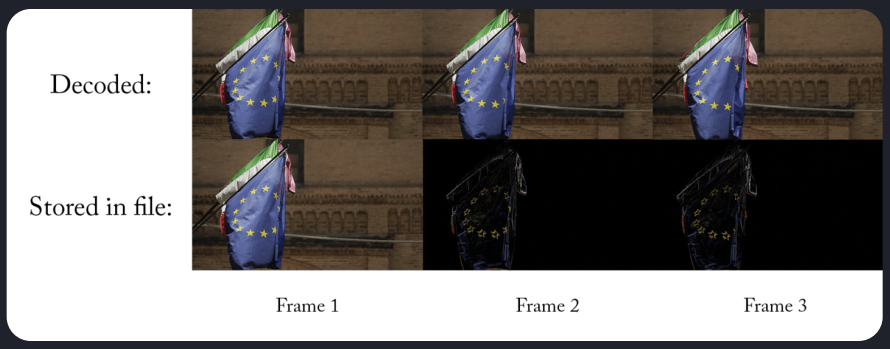
While it's almost always possible to import the camera files directly into the NLE and begin editing, it's seldom a good idea. Depending on what kind of camera and settings you used, it's very

likely that the codec from your camera is not well suited to editing video. If you are using a highly-compressed codec, if you are using a high-bitrate codec, or if you are recording in a log format, then an offline edit will probably allow you to work faster, on cheaper hardware.

## Highly-Compressed codecs can Slow Down Your Edi

Most lower to mid-range cameras record with codecs that use temporal compression, also known as long-GOP compression.

The simple explanation of a long-GOP is that, for each frame, the codec only captures what has changed between this frame and the previous frame. If the video doesn't include a lot of motion, then this means that the new file can be a LOT smaller than the original. The difference between this frame and the last frame is just a few pixels, so all you need to store is a few pixels. That's great!




The issue, however, is that these codecs tend only to work well when played forward. (If you're curious why, read our the deep dive on codecs). That's great for viewing on YouTube and your DVD player, but it's not great for editing, because when you're editing you're often jumping around, or playing a clip backward. It takes a lot more processing power to do those things quickly with a long-GOP codec. A high-end computer might have no trouble, but even a mid-range

computer will lag and stutter when you skim through the footage quickly or jump around.

Codecs that aren't long-GOP (a.k.a. Intra-frame codecs), however, can play backwards just as easily as forwards, and even a mid-low-end computer can skip around very smoothly. If you've only ever edited clips straight from the camera, you not might realize what you're missing.

The other thing that can cause issues with playback is raw video. Raw video needs to be converted before it can be displayed (sort of like a codec does), and some computers can't decode the raw file fast enough, especially if it's 4K. Ironically, both the low-end cameras and the highest-end cameras produce files that are hard to edit.

## High-Bitrate codecs can Slow Down Your Editing

For low to mid-range codecs, you don't have to worry about the bitrates at all. Once you start moving up the ladder, however, high bitrate codecs can cause issues with editing, especially if you're working on everyday computers.

The reason is that your computer needs to be able to read the data from your hard drive at a rate that is at least as high as your codec's bitrate. It makes sense – if your codec is 50Mb/s (fifty megabits per second), then your computer needs to be able to read that file from your hard drive at 50Mb/s, or else it'll fall behind and stutter.

[Note: Mb/s stands for megabits per second, while MB/s stands for megabytes per second. There are eight bits in a byte, so you need to multiply by 8 when converting from MB/s to Mb/s.]

The good news is that hard drives are getting faster every day, so 50Mb/s is never going to cause any problems. But what if you're editing ProRes 422HQ at 4K, which is 734Mb/s? The average external hard drive is only just barely fast enough to play that back, and some cheaper hard drives won't manage it.

And then, what if you're editing a multicam with three cameras? Now you need 3x that data rate: 2,202Mb/s. At that point, you're going to need to invest in some high-performance hard drives or RAIDs.

Here are some rough guidelines for common data storage speeds, though of course there will always be certain models that underperform or overperform.

Standard spinning drive: 100-120MB/s

Professional spinning drive: 150-200MB/s

Low-end RAID: 200-300 MB/s

Standard SSD: 400-500 MB/s

High-end RAID: 1000-2000 MB/s

High-end SSD: 2000-3000 MB/s

For some editors, working off of a high-performance storage system is not an issue. But there are significant savings both in cost and flexibility with using proxies. If you use proxies, it may be feasible to keep the entire film stored on the internal hard drive of a laptop, allowing you to work from the beach work from home, or from a plane or train.

Editing in log can slow down your editing

Shooting in log is a way of preserving as much of your dynamic range as possible, giving you more flexibility to achieve your desired look in post. Now that log profiles are available even on most prosumer cameras, it's an extremely popular way to work.

The downside is that the image that comes out of the camera doesn't look so great, so you need to add in a bunch of contrast and saturation in order to preview the image accurately.

This means that your editor will need to apply the appropriate LUT to all of the clips when editing. This can be annoying to manage, and it can also slow down the computer a bit, because it needs to first decode each frame and then apply the LUT before displaying it. It's possible to avoid that annoyance by editing the log footage without any LUT, but it's certainly not ideal.

The most common way to avoid this issue is by pre-baking the LUT into proxies. If you're going to convert your files into proxies before editing them, then you can apply the LUT during the transcode process. That way, the editor is always working with footage that has good contrast and color and never has to bother with LUTs. Note that you should only do this if you are using a Proxy workflow, not a Mezzanine workflow.

The offline editing process has so many benefits for the editors that it's nearly universal on feature films and TV shows. In our workflow analysis of the 2018 Oscar Nominees, we found that every single film used an offline workflow. And it's very common even on smaller projects as well. There are only a few cases where an offline edit isn't

a good idea.

1. When you have an extremely fast turnaround time. In those cases, it may not be acceptable to wait for proxy files to be generated, and so it can be faster to edit directly on the camera files.
2. When the files that you capture happen to be well-suited to editing. This is rare, though.
3. When you're going to be doing the editing and the color grading on a single computer.

Pitfalls of an Offline Workflow

The online-offline workflow is robust and efficient, but it has lots of moving parts. In practice, that means that you can fall into a number of technical and procedural traps, which will slow your work down considerably.

If you are using an online-offline workflow for your project, here are some pitfalls to avoid.

[Note, many of these examples are illustrated with DaVinci Resolve and Premiere Pro, but the principles should translate to any other NLE.]

## Pitfall #1: Not using timecode

Timecode is the glue that holds an online-offline workflow together. While Premiere Pro is relatively good at keeping track of clips that don't have timecode, Resolve more or less requires it. While Premiere Pro can assign a kind of "fake" timecode inside of a Premiere Pro project, it won't translate properly into Resolve via an XML.

A typical online-offline workflow involving Premiere Pro and

DaVinci Resolve consists of:

1. Preparing Proxies in Resolve
2. Doing the offline editing inside Premiere, with the timeline referencing offline media
3. Kicking out an XML from Premiere
4. Importing a folder of original digital negatives into a Resolve project
5. Relinking the imported timeline from the XML to the online clips inside the Resolve project

By default, without timecode, that timeline inside Resolve will have pretty much no way of identifying the proper in and out points for each clip in the timeline. Without timecode, Resolve might be able to place the clips into the sequence at the correct sequence in and out points, but there's no guarantee that the in and out points for the clips will be correct. Without timecode, each clip's in and out points within the sequence won't make it into Resolve. Without timecode, a clip's in and out points will be arbitrary.

Not all formats have timecode. The MP4 container, a very common container for H.264, is not even capable of containing timecode. This can be a bit confusing, because in many post-production apps themselves, the clips themselves can display a count that looks like timecode, even if it's not actually timecode. It's important to just be familiar with which formats can and cannot contain timecode.

Timecode is not just the particular count on your media that shows up in the Source Monitor. Just like your clip has a video stream composed of still images in a particular sequence, and an audio stream composed of audio samples in a particular sequence, any professional format that contains SMPTE-compliant timecode has an ancillary timecode stream that assigns each frame a particular hour, minute, second, and

frame, according to the SMPTE 12M-2 standard.

ProRes files wrapped in their native MOV container, or DNxHR files wrapped in MOV, MXF OP1a, or MXF OP-Atom, all contain timecode. Different camera manufacturers will list in their technical specifications whether or not their camera formats contain SMTPE-compliant timecode.

In Premiere, you can check the properties of a format in the QuickTime container, and it'll show you that there's a timecode stream.

As long as the proxies you transcode out of Resolve have timecode that matches the timecode in the original digital negatives from the camera, each post-production application should properly be able to switch back and forth between the online and offline formats as they're needed.

## Pitfall #2: Not having an organized folder structure

One of the most important parts of mastering an online-offline workflow is to make sure that the folder structures for the Original Camera Negatives (OCNs) and the proxies are identical, and that those folders are tightly organized.

The reasoning for having an organized folder structure is more profound than just keeping the online-offline workflow tamed. It's pertains to the same deeper reason that you're using an online-offline workflow in the first place. Having an organized folder structure and using an online-offline workflow are two methods that allow for many disparate collaborators to seamlessly work on a project quickly and efficiently.

Let's say you have two different collaborators that are separated by geography–this might be a director and editor who are in different cities, or two editors working on different deliverables, but who require the same source material.

If their folder structure is well-organized and identical, one collaborator can quickly direct the other to any particular shot. Both parties can recall the asset instantly. If different collaborators just start throwing their own assets into their own folder structures ad hoc, then projects will stall when collaborators waste time merely trying to recall particular assets.

## Not Having Unique Folder Names for Camera Rolls

It's important that a folder structure separates out every single camera memory card by a unique name. A common convention is to give every different camera on a particular project to get its own letter. We then count for each lettered camera, in the order that the footage was shot.

So, if we had three cameras, we'd call one camera A, one camera B, and one camera C. If we shot 5 cards on A, 2 on B, and 4 on C, the full list of camera cards would be:

- Camera A: A001, A002, A003, A004, and A005
- Camera B: B001 and B002
- Camera C: C001, C002, C003, and C004

This is one method to help alleviate the problem of camera files that might have name collisions. Even if you're not able to rename such problem files, having each camera card with its own unique name means that if you need to go digging into

an XML or a project file to check what the last good file path of a file might have been, you would quickly be able to see the difference between /A003/Clip001.mov and /F002/Clip001.mov.

## Not separating camera cards by date

For documentary work, wherein you might have directors or producers trying to remember something that they shot on a particular date, it's best not only to have each card separated out by a unique camera letter and number, but also by date.

So the folder structure at Freethink for a folder of camera originals and proxies has been standardized to look something like:

Original Digital Negatives

20180723

- 20180723_Sound
- A001
- A002
- B001
- B002
- C001

20180724

- A003
- A004

- B003
- B004
- C002

Dates are in ISO 8601 format, so that the filesystem lists them in order, chronologically, by default.

When a director is collaborating with an editor who wasn't on the shoot, the director can refer to footage in the natural way the human mind remembers it: "Hey, so we had a great shot of this on Tuesday morning…"

## Pitfall #3: Not waiting until picture lock to conform

The online-offline workflow is designed to optimize work for each individual kind of collaborator. The offline editor gets color-corrected intraframe files at a low data rate, synced to sound, so that they don't have to worry about syncing, color management, etc. The offline editor should just be able to play footage and set the order of the clips.

Broadly, the offline editor's deliverable is a recipe for how particular clips should be arranged on a timeline. That deliverable might technically be a .prproj file, or an XML, or an EDL, but the format doesn't matter. The goal is for the editor to hand off the particular recipe for the particular editing decisions they've made.

Online collaborators have very different goals, though. Colorists, VFX artists, sound editors, and sound designers aren't finessing the exact timing of particular cuts–in fact they're taking many precautions to ensure that the cuts on the piece they're working on don't change at all. It would be a

dire technical mistake for someone in the online editing process to make a timing change to any particular clip.

Given how online applications are designed, once a piece has been conformed for color grading and sound mixing, even minor tweaks to the timing of edits can quickly spiral into time-consuming headaches.

Imagine, for instance, if you've already conformed a picture-locked cut, and it's been sent out for color, VFX, and sound. Let's say a director or producer wants to just tweak something "simple," and just wants to swap one particular shot for another.

For the uninitiated, this seems like it would be extremely easy to do, because in an editing session, and editor could do this in a matter of seconds. However, if this piece has already gone to other collaborators, such a request creates a time-wasting headache compounded across multiple people. This headache would ensue for an assistant editor:

1. The director, editor, and/or producer would need to sit down and specify the exact sequence timecodes for what needs to change.
2. The AE would go grab the replacement clip and upload it.
3. The colorist would need to be informed and given the exact in and out points on the timeline.
4. The shot will need to be sent to the colorist—hopefully the colorist hadn't already spent a lot of time on that shot, because any such work will now have been for naught.
5. The VFX artist would need to be informed and given the exact in and out points on the timeline.
6. The shot would need to be sent to the VFX artist—hopefully the VFX artist hadn't already spent a lot of time on that shot, because any such work will now have been for naught
7. The sound editor would need to be informed and given the exact in and out points on the timeline.

8. The audio for the clip would need to be sent to the sound editor—hopefully the sound editor hadn't already spent a lot of time on that shot, because any such work will now have been for naught.

9. The sound designer would need to make sure that any sound design elements are swapped accordingly–hopefully the sound designer hadn't already spent a lot of time on that shot, because any such work will now have been for naught.

Are you seeing the pattern and problem? Making timing changes after picture lock is a great way to turn what would have been a 5-second task in the offline edit into up to an hour or more of wasted time across different collaborators.

Alexis Van Hurkman, author of the Color Correction Handbook and the DaVinci Resolve manual, says in his book that "locking the edit should not be looked at as a technological limitation but as a scheduling milestone. Sooner or later, the director and producer will have to make up their minds, be done with the edit, and allow the project to go through finishing.

## Pitfall #4: Not having clips with unique file names

When swapping the offline media in Premiere with the online media into DaVinci Resolve, the filename itself is one of the primary ways that the app relinks to a clip. As the application traverses a specified directory recursively, it's looking for particular filename matches.

Now, many "prosumer" cameras like DSLRs, GoPros, and drones simply aren't able to keep track of clip numbering across card changes. In professional RED, Canon, Sony, and Arri cameras, the numbering of a new card will pick up from whatever the last clip of the previous card was numbered; but in a "prosumer" camera, every time a card is inserted and formatted, the numbering restarts. This could result in a single

day's worth of footage that has 10 different clips all named "C000," 10 clips named "C001," 10 clips named "C002," et cetera.

One way to avoid this would be to stick to cameras that can keep track of numbering across cards,–but with tight budgets, that's not always possible.

Another solution would be to simply rename the clips so that every single one in the entire project is unique. Sony Catalyst Prepare can rename camera files as it ingests, while properly keeping all the sidecar metadata intact for Sony formats. The Mac OS Finder has a handy built-in function to batch rename files.

## Pitfall #5: Not using the right offline format

If you're going to go through the whole process of preparing proxies for the offline editor, wherein you're syncing audio, color correcting from log to Rec. 709, and meticulously keeping your folder structure for the proxies identical to the folder structure of the camera originals, you might as well use the correct format.

One of the primary reasons to use proxies in the offline edit is to make those proxies portable and easy to work with. Selecting a proper format designed for this is crucial.

A good proxy format accomplishes two objectives:

1. Relatively low data rate
2. Easy to serve up frames–typically by using only intraframe compression and entirely avoiding interframe compression

A format with a low data rate will allow for more footage on

fewer hard drives, and intraframe compression will allow the footage to be played on low-powered systems without very powerful CPUs or GPUs.

Remember, these proxies are just for the editor to figure out the timing of the edits; the format is not meant to preserve the image fidelity. Preserving image fidelity will be handled later when conforming the cut to the camera originals or a suitable digital intermediate format in preparation for the final color grade.

Two good options for proxies are Apple ProRes 422 Proxy or DNxHR LB.

## Pitfall #6: Not using data burn-ins

The tightly-organized folder structure is one part of the workflow that allows for quick recall among different collaborators, but it's not the only part. Data burn-ins on the proxies are also crucial. (Data burn-in's are the various metadata and/or timecode information related to a clip burned onto the image.)

The folder structure discussed above can allow a collaborator to identify particular shots, camera cards, and shoot days, but it's also important to include data burn-ins on those proxies as well.

Freethink's practice is to include data burn-ins on the proxies that include:

- Clip name
- SRC TC [Resolve's term for the embedded timecode of the video track]

- AUD TC [the embedded timecode of the audio track, in case the production sound mixer didn't jam sync perfectly, or if something went wrong with the timecode in the camera]
- Date [the date the clip was shot]
- Card [the camera card assigned, i.e. "A001"]

With data burn-ins on all the proxies, any collaborator can quickly call up a shot in the Source Monitor, and within seconds, navigate to any frame in a whole project. This scales down to a short commercial that might only have have an hour of footage all the way up to a documentary that might have hundreds of hours of footage.

## Pitfall #7: Not understanding sizing between the online and offline apps

For many editors, it can be confusing to understand the relationship between how different programs treat the reframing of clips. Pulling an XML from Premiere Pro into DaVinci Resolve can be frustrating, because the resulting positioning data is often wildly inconsistent; some shots will match up perfectly, but others will be way too small, way too big, or be placed completely wrong within the frame.

Turns out there is a secret. There are particular combinations of settings across both applications, which, if you rigorously adhere to, will allow you to consistently and reliably transfer your repositioning data out of Premiere Pro and into DaVinci Resolve via the XML.

Instead of spending hours and hours going shot by shot in Resolve and fixing positioning errors, you can now trust your shots to come into Resolve with proper placement.

This will obviously vary if you're using other NLEs like FCP X or Media Composer, but the lesson here is that you can, and should, systematically run your own experiments so that you can figure out a reliable combinations of settings that work for you.

## Pitfall #8: Not syncing audio losslessly

Recall that the purpose of the proxy workflow is organizational –to allow for each individual collaborator to get exactly what they need to get to work creatively. How does audio fit into this?

For the process of syncing audio and then rendering your proxies, it's best practice to take the interleaved [a.k.a. polyphonic] WAV files from the production sound mixer and losslessly rewrap the files into the proxies.

It's tempting to transcode to a lossy, compressed audio format like AAC, but it doesn't really accomplish anything. By losslessly rewrapping the uncompressed audio, not only will the offline editor have access to each individual microphone's own ISO track, but when it comes time to conform the sound files for sound mixing and sound design, there won't be any additional work in tracking down the original files from the production sound mixer's field recorder.

Relative to video files, audio files–even uncompressed audio files– are tiny, so transcoding into a lossy format like AAC serves no purpose other than to create pointless additional work and to waste time. If you've already set up your storage to be able to handle video files, you should still have plenty of storage space for uncompressed audio.

Within the Avid suite, there are indeed tools to go fetch the

original field recorder files, but if you just include all of the uncompressed iso tracks, you can skip that step altogether.

## Conclusion

The online-offline workflow, when properly implemented, is robust and has been serving filmmakers well for about a century. The online-offline workflow enables collaboration and gains from the division of labor. However, in the modern age, with file-based workflows, we have to keep track of much more than physical reels of film. As you craft and refine the online-offline workflow for your next project, heed these warnings, and don't fall prey to these mistakes. Your collaborators involved in the process will thank you for keeping everything running smoothly and efficiently.

Contributed by Seth Goldin, Tech, post-production, and workflow at Freethink Media.

## All-Online Editing

On a large project like a feature film or TV show, for the most part, the various roles in post-production are kept fairly separate. The editors just edit. The colorist just colors. The sound mixer just mixes sound. On smaller projects, however, the roles are much more fluid, and it's common for a single person to do the jobs of colorist, editor, sound mixer, etc.

In that case, it's more common to skip the offline edit because you lose some of the benefits. The biggest benefit of the offline workflow is that it allows you to edit smoothly on cheaper hardware. Among 2018's Oscar nominees, *Darkest Hour* and *The Shape of*

*Water* both used old style Mac Pro towers that were far behind the latest technology.

With an offline edit, the editors are able to work with much cheaper computers (or even laptops), and then they can hand off the project to the colorist whose high-horsepower machine can handle the full resolution and quality of the original camera files.

"Baby Driver" lead editor Paul Machliss editing on-set. Image courtesy Paul Machliss.

If the entire process (edit, color, sound, VFX) is happening on a single machine, however, then that machine must be powerful enough to work with the original camera footage. So, if you already have high-performance storage and high-performance computing, then you probably won't see as much of a speed boost when switching to proxies. Of course, it's very easy to find out.

There is also a kind of hybrid All-Online approach that we call a Mezzanine workflow. It provides some of the advantages of online editing and also removes the need to relink back to the original camera files. This workflow is becoming less common, however.

Deep Dive: Mezzanine workflow

A mezzanine workflow is a kind of offline workflow that can provide a smoother editing process than editing directly from the camera files.

The most common offline workflow is a proxy workflow, where you use smaller, low-quality files to edit with and then reconnect back to the camera files afterward.

With a mezzanine workflow, you transcode your camera files into a codec that is both good for editing and very high-quality (not very lossy). Because the codec is very high quality, almost all of the original information from the camera files has been preserved, and so it's not necessary to re-link back to the camera files – you can just export directly from the intermediate files or hand them off to your color correction team.

There will be some theoretical loss of information when you transcode into this codec, but if you pick a good enough intermediate codec, the difference will be small enough that you don't need to worry about it.

## The proxy workflow sounded pretty good. Why do the Mezzanine workflow?

Part of the reason why the Mezzanine workflow is common is because it used to be a lot harder to use a proxy workflow. Some of the major software providers didn't make it particularly easy to relink back to the original camera files, and so people would choose a Mezzanine workflow for the simplicity. Nowadays, however, the offline/online switch is pretty easy to do in any editing package.

One exception could be when you have a lot of mixed

footage types. If you have multiple frame rates and frame sizes in the same project, switching back and forth from the proxies to the capture codecs can be a headache. A mezzanine workflow solves that problem.

If you are using some third-party tools to help prep and organize your footage before you start cutting, those can also make the relinking process more tricky with a proxy workflow. One common example might be software that automatically syncs audio tracks or multicam shoots.

One downside to the mezzanine workflow, however, is that you can't "bake in" the LUTs for your editor – you're going to need to apply a LUT via a color-correction effect in your editing software. If you were to include the LUT in your transcode for Mezzanine workflow, you would be losing all of the benefits of recording in log in the first place.

The other obvious downside is that you need to store all of these (much larger) files while you're editing, and you will also need higher-speed storage than you'd need with a proxy workflow.

## How to pick a Mezzanine codec

The key to picking a good Mezzanine codec is to make sure that you are preserving all of the information from your capture codec. An intermediate codec will never make your images better (more detailed explanation below), but it can definitely make them worse if you choose the wrong codec. The important thing is to understand the details of your original footage and make sure that your intermediate codec is at least as good as your capture codec in each area. If you capture your footage on a DSLR like a Sony A7Sii at 4K, then you will be recording in a 4:2:0, 8-bit, Long-GOP codec at 100Mbps. You want an intermediate codec that is at least

4:2:0 and 8-bit. Going beyond these values (e.g. to 4:4:4 and 12-bit) won't hurt, but it also won't help at all, so it's probably not worth the extra storage space.

An intermediate codec will never make your images better

Transcoding Footage to a Higher Bit Depth Won't Increase Quality

Let's say, for example, that we want to go with a ProRes codec. We have 4 options to choose from that are 4:2:2 and 10-bit.

145Mb/s ProRes 422 Proxy

328Mb/s ProRes 422 LT

471Mb/s ProRes 422

707Mb/s ProRes 422 HQ

You might think that all you need is to match the camera bitrate (100Mbps), but you actually need to greatly exceed the camera bitrate. This is because h.264 is a much more efficient codec than ProRes. Because h.264 uses long-GOP compression, it can pack a lot more information into those 100 megabits than ProRes can. In order for ProRes to match the image quality of h.264, you need a much higher bitrate. We would recommend only using ProRes 422 or ProRes 422 HQ if you're starting with a 100Mbps h.264 codec. ProRes 422 will probably do just fine, but if you have lots of storage space, then going up to ProRes 422 HQ will have a slight edge.

While it's fine to simply match the bit-depth and color sampling when choosing an intermediate, you should always increase the bitrate at least a little. If you're going from long-GOP to a non-long GOP codec, then you should increase the

bitrate a lot.

Side note: If you wanted to go with DNxHD instead of ProRes, you have similar options, except that DNxHD also offers an 8-bit version for the lower-end codecs. Since our footage is 8-bit to start with, that won't hurt us at all.

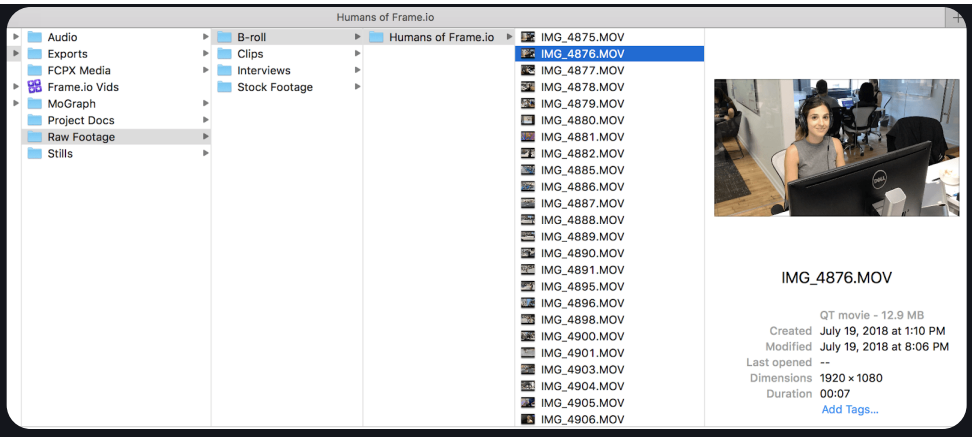
## Project Organization (Logging)

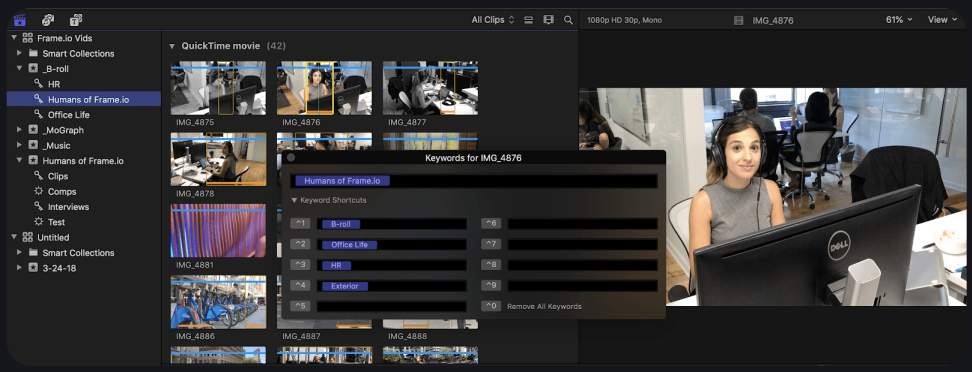
Once the assistants have copied the new set of dailies to the central shared storage, they will begin organizing them in the NLE (editing software) according to the system that the lead editor prefers.

Although the media files for a film tend to be very large, the project files created by the NLE are usually quite small. The project file contains pointers to all of the media files, but it does not hold those media files themselves–they are stored separately. The media files for a 30-second advertisement might be in the hundreds of gigabytes, while the project file might be only 5 megabytes.

(Note: if you're a Final Cut Pro X user, you have the option to ingest media directly into the Library, which is Final Cut Pro's version of a "project file." In those circumstance, the Library will balloon to the size of the content you import, in addition to the various files it creates. This workflow is uncommon, though).

In the simplest scenario, the file structure inside the NLE will exactly match the structure of the files as they are stored in the shared storage. That simplicity can be nice, but on the other hand it may require moving a lot of files around on the data server.




…the keyword or bin structure within the NLE.

Alternatively, it's common to leave the files in exactly the same folder structure in which they arrived but organize them differently inside the NLE.

## Deep dive: Bin organization

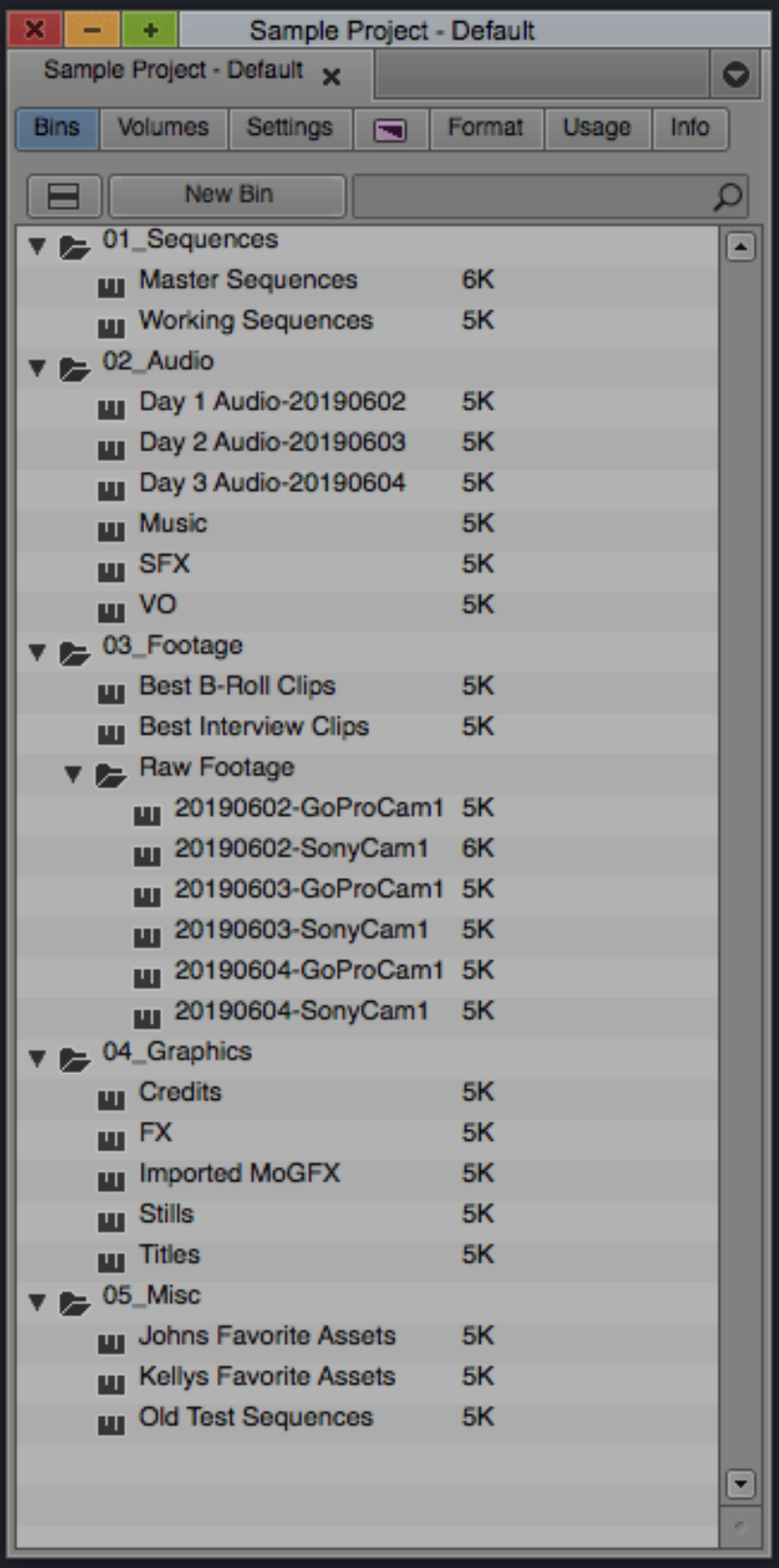
Proper bin organization allows any editor stepping into a project to know exactly where every element of the project is without having lengthy searches. This is done by having a clear, consistent bin organization system.

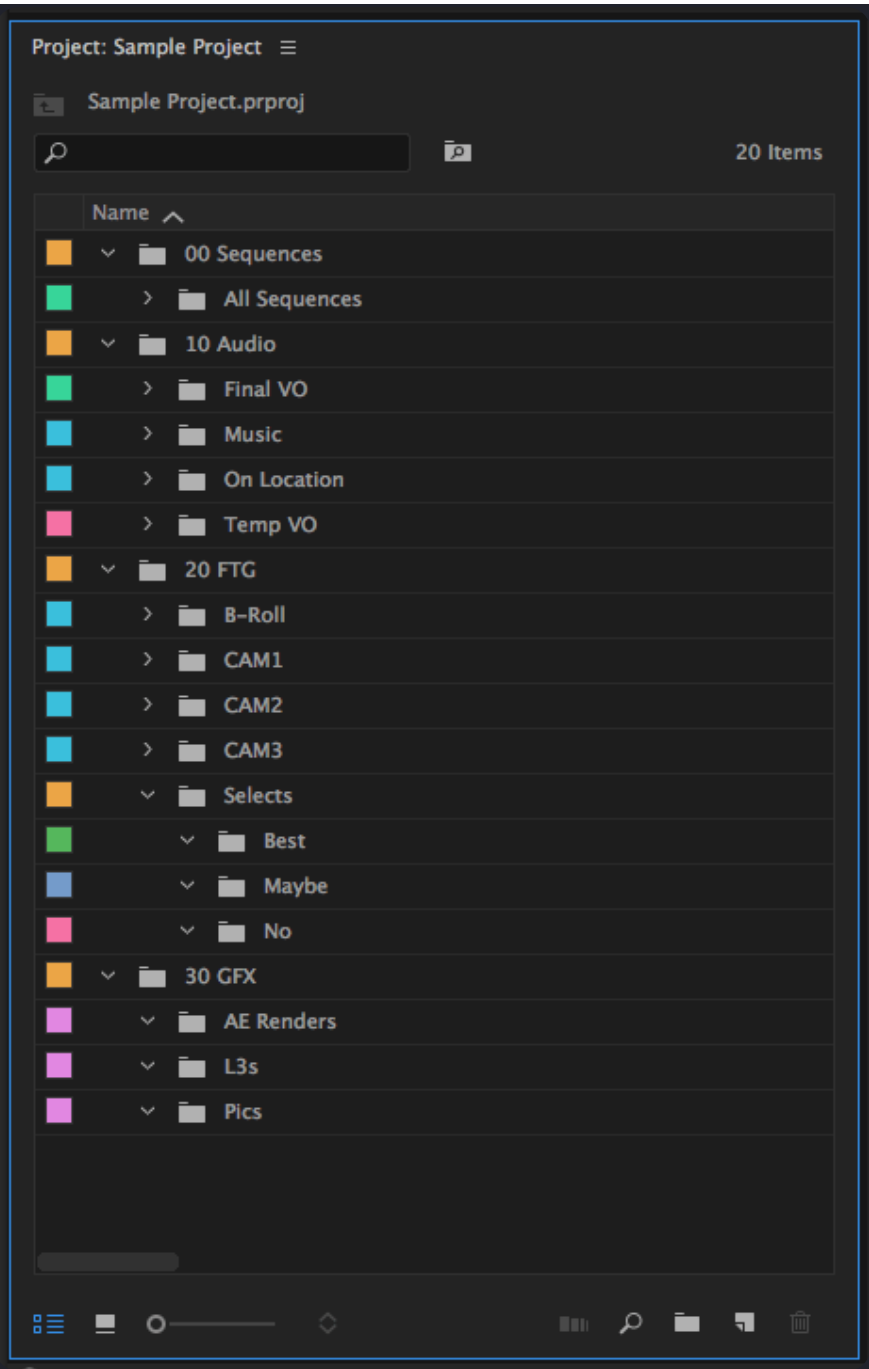
The bin organization system can differ from post house to post house, but they all revolve around a couple key strategies.

The first strategy is to keep similar assets together. There are generally five categories of assets – sequences, audio, footage, graphics, and miscellaneous. These categories should get their own folder in the NLE. Inside that folder are the numerous bins associated with that category. Bins in the audio category would include music, sound effects (commonly written as SFX), voice-over (commonly written as VO), and any on-location audio that was recorded separately from the footage. For example, if there were three days of shooting there would be bins labeled "Day 1 Audio," "Day 2 Audio," and "Day 3 Audio." These bins would more than likely have the date of the shoot in the name of the bin as well.

These categories of assets are usually written with numbers in front of them. It's common to see folders labeled something like, "00_Sequences," "10_Audio," "20_Footage," "30_Graphics," and "40_Misc." The numbers are for labeling purposes only, to force the NLE to order them that way when sorted alphabetically.

Here are two sample bin organization setups:


Sample Project - Default
Sample Project - Default
Bins
Volumes
Settings
Format
Usage
Info
New Bin
01_Sequences
Master Sequences 6K
Working Sequences 5K
02_Audio
Day 1 Audio-20190602 5K
Day 2 Audio-20190603 5K
Day 3 Audio-20190604 5K
Music 5K
SFX 5K
VO 5K
03_Footage
Best B-Roll Clips 5K
Best Interview Clips 5K
Raw Footage
20190602-GoProCam1 5K
20190602-SonyCam1 6K
20190603-GoProCam1 5K
20190603-SonyCam1 5K
20190604-GoProCam1 5K
20190604-SonyCam1 5K
04_Graphics
Credits 5K
FX 5K
Imported MoGFX 5K
Stills 5K
Titles 5K
05_Misc
Johns Favorite Assets 5K
Kellys Favorite Assets 5K
Old Test Sequences 5K

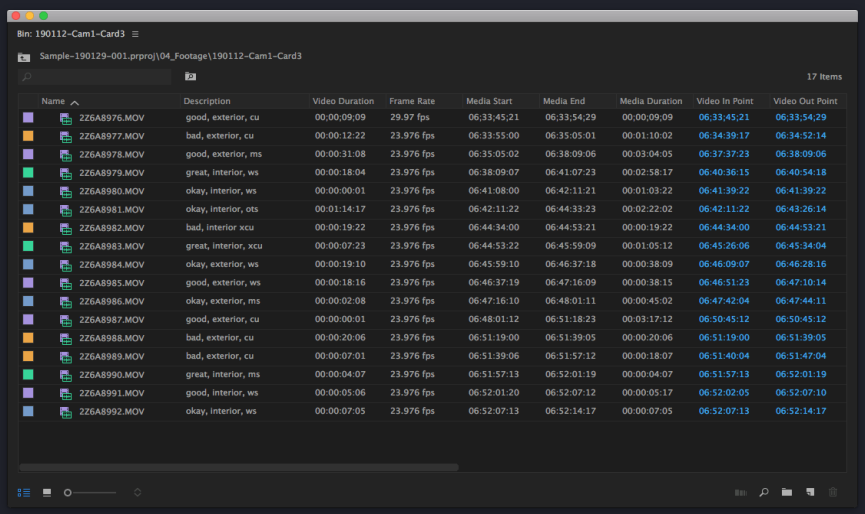



Footage bins usually require a bit more attention. Depending on the project, a common strategy is to tag, rate, color or somehow label clips inside footage bins after they have been imported into the project. This strategy separates the good clips from the bad clips and the great clips from the good clips.

Editors or assistant editors will evaluate the clips and use one of those labeling methods so they can quickly find usable

clips. Finding clips is done by sifting, searching or sorting columns inside of a bin. If a 10-second shot is needed a bin can be sifted to just show the best shots then sorted by duration so all that's left is shots that fit the criteria. This can be narrowed down even further with more metadata associated with footage. This shot also needs to be an exterior shot? With proper labeling this is no problem – just add another element to the search criteria.

Here's an example of a footage bin that's been color-coded with information added in the Description column:



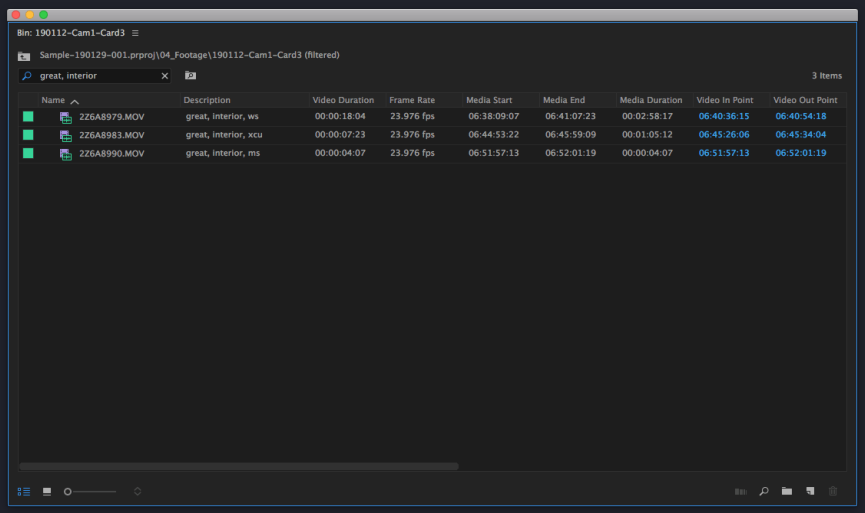
Bin: 190112-Cam1-Card3

Sample-190129-001.prproj\04_Footage\190112-Cam1-Card3

17 Items

| Name | Description | Video Duration | Frame Rate | Media Start | Media End | Media Duration | Video In Point | Video Out Point |
|---|---|---|---|---|---|---|---|---|
| 2Z6A8976.MOV | good, exterior, cu | 00;00;09;09 | 29.97 fps | 06;33;45;21 | 06;33;54;29 | 00;00;09;09 | 06;33;45;21 | 06;33;54;29 |
| 2Z6A8977.MOV | bad, exterior, cu | 00:00:12:22 | 23.976 fps | 06:33:55:00 | 06:35:05:01 | 00:01:10:02 | 06:34:39:17 | 06:34:52:14 |
| 2Z6A8978.MOV | good, exterior, ms | 00:00:31:08 | 23.976 fps | 06:35:05:02 | 06:38:09:06 | 00:03:04:05 | 06:37:37:23 | 06:38:09:06 |
| 2Z6A8979.MOV | great, interior, ws | 00:00:18:04 | 23.976 fps | 06:38:09:07 | 06:41:07:23 | 00:02:58:17 | 06:40:36:15 | 06:40:54:18 |
| 2Z6A8980.MOV | okay, interior, ws | 00:00:00:01 | 23.976 fps | 06:41:08:00 | 06:42:11:21 | 00:01:03:22 | 06:41:39:22 | 06:41:39:22 |
| 2Z6A8981.MOV | okay, interior, ots | 00:01:14:17 | 23.976 fps | 06:42:11:22 | 06:44:33:23 | 00:02:22:02 | 06:42:11:22 | 06:43:26:14 |
| 2Z6A8982.MOV | bad, interior xcu | 00:00:19:22 | 23.976 fps | 06:44:34:00 | 06:44:53:21 | 00:00:19:22 | 06:44:34:00 | 06:44:53:21 |
| 2Z6A8983.MOV | great, interior, xcu | 00:00:07:23 | 23.976 fps | 06:44:53:22 | 06:45:59:09 | 00:01:05:12 | 06:45:26:06 | 06:45:34:04 |
| 2Z6A8984.MOV | okay, exterior, ws | 00:00:19:10 | 23.976 fps | 06:45:59:10 | 06:46:37:18 | 00:00:38:09 | 06:46:09:07 | 06:46:28:16 |
| 2Z6A8985.MOV | good, exterior, ws | 00:00:18:16 | 23.976 fps | 06:46:37:19 | 06:47:16:09 | 00:00:38:15 | 06:46:51:23 | 06:47:10:14 |
| 2Z6A8986.MOV | okay, exterior, ms | 00:00:02:08 | 23.976 fps | 06:47:16:10 | 06:48:01:11 | 00:00:45:02 | 06:47:42:04 | 06:47:44:11 |
| 2Z6A8987.MOV | good, exterior, cu | 00:00:00:01 | 23.976 fps | 06:48:01:12 | 06:51:18:23 | 00:03:17:12 | 06:50:45:12 | 06:50:45:12 |
| 2Z6A8988.MOV | bad, exterior, cu | 00:00:20:06 | 23.976 fps | 06:51:19:00 | 06:51:39:05 | 00:00:20:06 | 06:51:19:00 | 06:51:39:05 |
| 2Z6A8989.MOV | bad, exterior, cu | 00:00:07:01 | 23.976 fps | 06:51:39:06 | 06:51:57:12 | 00:00:18:07 | 06:51:40:04 | 06:51:47:04 |
| 2Z6A8990.MOV | great, interior, ms | 00:00:04:07 | 23.976 fps | 06:51:57:13 | 06:52:01:19 | 00:00:04:07 | 06:51:57:13 | 06:52:01:19 |
| 2Z6A8991.MOV | good, interior, ws | 00:00:05:06 | 23.976 fps | 06:52:01:20 | 06:52:07:12 | 00:00:05:17 | 06:52:02:05 | 06:52:07:10 |
| 2Z6A8992.MOV | okay, interior, ws | 00:00:07:05 | 23.976 fps | 06:52:07:13 | 06:52:14:17 | 00:00:07:05 | 06:52:07:13 | 06:52:14:17 |

Here's that same bin filtered to show just the "great interior" shots:



Bin: 190112-Cam1-Card3

Sample-190129-001.prproj\04_Footage\190112-Cam1-Card3 (filtered)

great, interior

3 Items

| Name | Description | Video Duration | Frame Rate | Media Start | Media End | Media Duration | Video In Point | Video Out Point |
|---|---|---|---|---|---|---|---|---|
| 2Z6A8979.MOV | great, interior, ws | 00:00:18:04 | 23.976 fps | 06:38:09:07 | 06:41:07:23 | 00:02:58:17 | 06:40:36:15 | 06:40:54:18 |
| 2Z6A8983.MOV | great, interior, xcu | 00:00:07:23 | 23.976 fps | 06:44:53:22 | 06:45:59:09 | 00:01:05:12 | 06:45:26:06 | 06:45:34:04 |
| 2Z6A8990.MOV | great, interior, ms | 00:00:04:07 | 23.976 fps | 06:51:57:13 | 06:52:01:19 | 00:00:04:07 | 06:51:57:13 | 06:52:01:19 |

Some productions use shared bins across multiple projects. Let's say four episodes of a TV show are being edited at the same time by four different editors in four different projects. All four editors can share resources from a shared project containing elements like the opening title sequence and graphics packages used throughout the series. Those bins live in a different project from each episode's project but can be opened and used by all the editors. See bin locking and simultaneous access for more information.

The importance of bin and asset organization can often be overlooked. Once the cameras stop rolling all eyes turn towards the editor to start putting together the first cut. However there needs to be proper time spent at the beginning of the project to organize the bins in order for the editor to be efficient throughout the entire post-production process.

Contributed by Josh Short, Edit Video Faster.

If the editor wants to make heavy use of metadata in the editing process, the assistants will also tag and categorize the new clips to help with the editing process later.

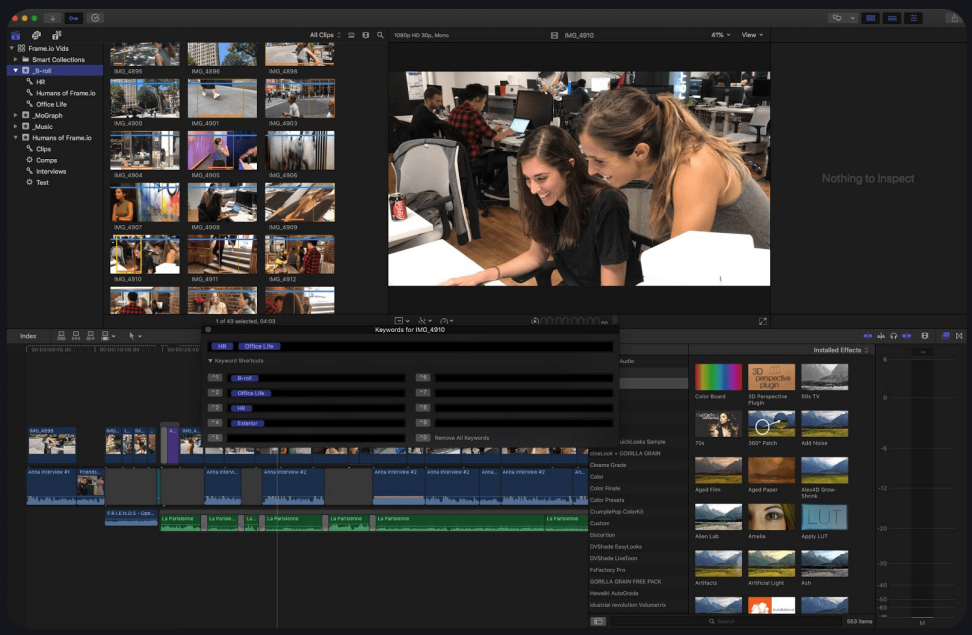
Earlier, we mentioned the metadata that is created during the shoot (camera and lens information, director's notes, etc.), but some editors use more detailed metadata to organize their edit.

A traditional hierarchical project organization may begin with the major sections of the film (acts), and then within each act will be a folder for each sequence, and then all of the shots in the sequence, and finally all of the takes of that shot.

With Final Cut Pro X, Apple created a new approach to editing in the last decade that uses a different paradigm for organizing

footage. In addition to thinking about the footage hierarchically, the editor also thinks in terms of the attributes of each clip. Using attributes (or "tags"), the editor can instantly view all of the close-ups of a particular character in one screen, without having to go through all of the different folders.

This approach can be extremely powerful especially for projects that don't have a strictly planned-out linear script, like an ad or a music video.




In the years since FCP X was released, the other NLEs have all added some sort of tool for metadata organization, though none is as sophisticated as FCP X.

Metadata-driven editing

## What is Metadata?

Metadata is extra information attached to your source files, which is used to describe those files in finer detail.

For example, have you ever opened a shot from your phone in your favorite image editor? If you head to the Properties or File Info dialog, you'll see some interesting metadata. Your phone's make and model, the shot's exposure and shutter speed, even the location where the photos were taken – all of this extra info is seamlessly folded into every shot on your phone. That's metadata.

But how does metadata affect postproduction? Let's sift it into two categories:

1. Useful Metadata for Systems
2. Meaningful Metadata for Humans

## Useful for Systems

Systems use metadata to process files in granular ways. For instance, operating systems use metadata to display common file attributes such as Type, Size, and Date Modified, which can be visually grouped and sorted with the click of a column.

NLE's and VFX tools also use metadata to expose crucial details about your project's assets. Is this clip a QuickTime MOV, or MP4? Is the codec a flavor of ProRes, or H.264? What's the frame rate? Bitrate? Does it have Linear PCM 48 KHz stereo, or no sound at all? All of this metadata is baked into your files to help *this* system, the host application, do its job.

## Meaningful for Humans

But not all metadata is for machine-use only. More and more

postproduction software environments give you, the human, the tools necessary for applying and wielding metadata in a meaningful way.

## Keywords

Keywords are specific terms or phrases you can attach to each clip. Once you've created your own keyword set, it can help you and your team get (and keep) your arms around your project. The good news? You don't need an exhaustive set of keywords to see the payoff.

For example, keywords can help you stay organized. Some apps generate keywords based on your project's folder structure when you import media. It's a rudimentary method of asset tracking, but if you're following your project's file and folder naming convention, it's another breadcrumb trail for your assets' source of origin. This comes in handy when working with incoming assets from disparate sources.

Keywords also help you look at your footage as ideas, not just as clips in a bin. Names, places, emotions, moments – all of these ideas are candidates for keywords that can quickly become part of your team's vocabulary.

Once those keywords are available, that metadata becomes the basis for locating any asset in your project. Type some keywords into your NLE or VFX app's Search function to quickly narrow down the number of clips. Don't know any of the keywords? Some apps allow you to filter results by selecting keywords with checkboxes and logical conditions. Others let you save those keyword searches or filters for later use. In other words, adding keywords lets you create a custom search engine for your project!

So do you have *Interview* shots of *John Smith*, but also shots of John in the *Factory* as *B-Roll*? Isolating those ideas, then

translating them into keywords, will help you find what you need even faster during the edit.

### Project-Specific Information

Project-specific metadata may also come from other sources like cameras or Script Supervisors, in the form of:

- Reel
- Scene
- Camera make and model
- Camera angle
- Crew and personnel names
- Circle takes (takes that the director liked on set)
- Continuity notes

If available, these become another way to quickly find and filter project assets so you can keep that momentum going during the edit.

Metadata – useful for systems, but can be meaningful for you. All postproduction apps support system-grade metadata, but more and more are supporting meaningful metadata which allows you to organize and sift through hours of footage in mere seconds. When used effectively, it can help you make creative decisions more efficiently, collaborate successfully, and deliver on time.

## Editing with Metadata

When you use an NLE that is very highly metadata-oriented,

and you make heavy use of its metadata features, it can open up an entirely different way of organizing and selecting your footage.

Organizing clips and folders is one of the most important tasks in editing because, if you do it properly, it allows you to find the clips that you need much more quickly. Traditional NLEs have a whole suite of features that allow you to identify the right clips quickly.

- Hierarchical Bins. Many editors organize their clips into a series of nested bins (a.k.a. folders). The level may be three bins, one for each act of the film. Inside each act is a bin for each scene, and inside each scene is a bin for each camera, and inside each of those is a group of clips.
- Subclips. A subclip is a portion of a clip that is saved as its own clip. You may have a 3-minute-long take that's mostly not great, but there's one particular line that's beautifully spoken. The editor might create a subclip of just that one line.
- Stringouts. A stringout of a scene is just a sequence that includes every single shot from the scene laid out in a row. The editor can then watch through and choose the best ones.
- Selects. A selects sequence is a collection of just the best clips from a scene. It's common to start with a stringout sequence, make a copy, and then delete all of the not-as-good takes. What's left are your selects.

Metadata-driven editing replaces all four of those techniques with two simple but powerful functions: tagging, and searching.

## Tagging

Tagging happens at the beginning of the project, and it's as

simple as it sounds. The editor (or an assistant) tags each clip with all of the attributes that may be useful later. Some software calls these "keywords", but the concept is the same.

A clip will be tagged with the obvious information like the act number, the scene number, the camera, etc. In fact, those may automatically be applied if the production crew has handled their metadata properly.

But there is also a lot of more interesting metadata to add. You might add the names of the characters in the shot, whether the director was happy with the performance, the type of shot (wide, medium, close-up), whether the shot is moving or not, whether a particular prop is visible in the shot, or whether or not the actor remembered to wear his vest.

You can also add metadata to a portion of a clip, instead of to the whole clip. In the above scenario where a traditional editor might make a subclip, you would mark that same section as "good."

Some of this metadata can even be added automatically by computers using machine-learning technology to automatically detect characteristics of the shot.

## Searching

Now that you've put in all of the work to tag your footage with metadata, it's time to make use of it. "Collecting" is just a general term for how you work with metadata, and it's as simple as typing a few words into a search bar.

Typing "4 Rachel" will instantly show you all of the shots of Rachel from scene 4. If you add "best", then it will only show you the takes that you've also marked as the best ones.

Perhaps Rachel begins the scene by sitting on the couch and then walks to the window. If you add "couch", that show you the best shots that include both Rachel and the couch in scene 4. You did all of that just by typing "4 Rachel best couch".

Now that you've collected the shots you wanted, you can save this view for later, but that is often not necessary, since it only takes a couple of seconds to type those same words again.

If you want a stringout of scene 4, all you do is type "4" into the search bar. If you want your selects of scene 4, you just type "4 best".

As you can see, metadata-driven editing tends to be much more fluid and adaptable than traditional organizational methods, but it requires the editorial team to shift how they think about sorting and finding clips, which can be disruptive to existing workflows.

FCP X was the first NLE to provide advanced metadata features, but metadata-driven editing is gaining in popularity and many different tools are beginning to add some similar features.

Contributed by Isaac Terronez, Producer, video editor, and technologist.

## LUTs in the edit

Hopefully, at this point the basic color grade has been "baked into" the images by the DIT or Dailies team, which means that no extra color information needs to be included in the edit stage. If the dailies have arrived in a log format, then the assistant editors will

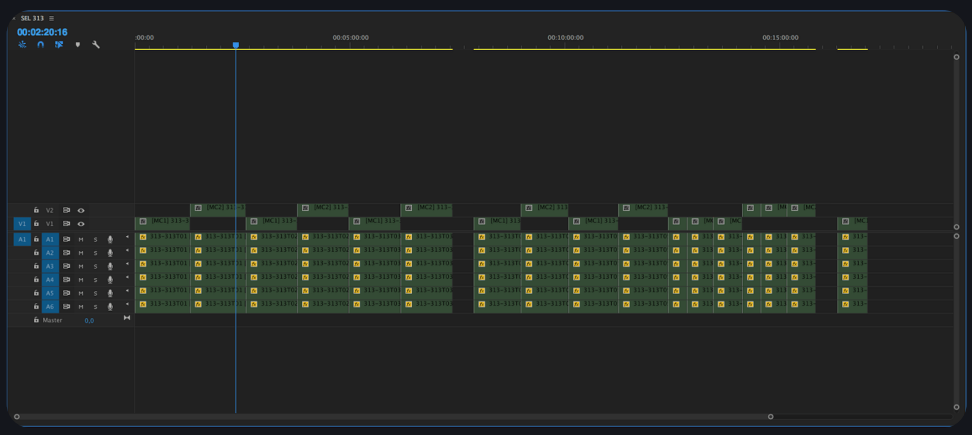
need to manage the LUTs or CDLs and make sure that they are properly applied to all media, so that the editor can properly judge each shot.

Some NLEs prefer to apply the LUTs at the clip level, while others prefer to apply them at the timeline level. Either is usually possible, but it's important to choose one approach.

## Stringouts

Once the clips are organized and/or tagged, the assistants often build stringouts for each scene, where every single take from the scene is laid out on a single timeline.

Then, the editor will begin working by watching straight through each take and making notes, often by placing markers directly on the timeline or by moving their favorite takes to a separate track.




These long sequences serve as another form of organization. If the editor wants to find a particular close-up of Bob from Scene 14, they can look in the Scene 14 bin, or they can look through the stringout sequence.

Deep Dive: Stringouts

A perfect edit is like the perfect outfit. Every shot goes together seamlessly and feels like it was always meant to be exactly as it is, like a perfectly-coordinated outfit. Of course, with that perfect outfit, you aren't seeing the entire piece of clothing: a shirt is tucked into pants, partially hidden behind a jacket or a scarf. The only time you see all the items in a wardrobe, the ones that go together and the ones that don't, is on laundry day, when they all get hung up on the line one at a time, with plenty of space between them.

That's a stringout. It's any time in the post process where you put together a large selection of footage in order to see it all at once and know what your options are. Back in the film production days, you got a lab roll of dailies, scanned it in and had a natural stringout that post teams could watch through. With digital production, however, you get a bin full of clips, and stringouts are the main tool used for getting a handle on precisely what footage was shot.

Because of the multi-faceted nature of post-production, there are many different situations where an assistant might be expected to prepare a stringout for different types of material, and it's good to have a handle on all of them.

## Narrative Stringouts

The most common stringout is a dailies stringout that shows you all of your shots for a given scene in order. Generally the assistant, or sometimes the lead editor, will trim the top and tails of a scene to only include the "meat" and not the slates, since often slate information is available in the audio file or the clip name and you don't need to see the slate in the picture in the string.

However, be careful when cutting off top and tails that you

don't trim useable material. There are many performances built out of the moments after the slate leaves frame and before the director calls "action." At the stringout stage, trim out parts of the shot that are clearly unusable (if there is a camera assistant holding a slate at the start, or after cut if the camera operator pans over to see the boom operator), but leave as much room as possible. At this stage, it's still not possible to know precisely what parts of the shot will be used.

If multiple cameras are used, some editors are tempted to use Multicam mode and to only watch one angle of the dailies. While Multicam mode might be useful for syncing with preferred audio and as an editing tool, at the stringout phase it's generally useful to lay out each camera one at a time. Watching through each angle, while it might occasionally be tedious, will offer real depth of knowledge for the material.

## Dialogue Stringouts

A slightly less common, and more laborious, type of stringout is a dialogue stringout. This is the creation of a new timeline for every single line of dialogue in the script, which then cuts together every single angle on that line of dialogue. If a line shows up in the wide shot, a medium, a close-up, and an over the shoulder, even if it's not focused on the key actor, you include it in this stringout.

For many first-time assistants, this can seem like incredibly tedious work. Especially since, while most top editors pride themselves on watching all of the footage, they generally watch every take in the normal way, from beginning to end, making notes. It's rare that every dialogue stringout is watched from beginning to end.

This stringout is essential prep work that is designed to front-load effort to save valuable time later. Its biggest benefit is

when the director, producer, or other client is in the suite. This in-person time is becoming increasingly valuable and hard to get, and editors want to spend as much of that time as possible on productive, creative work.

In those sessions, it's very common to have a director say "I'm not sure that's our best take on that line, can I see alternates?" In that moment, if you don't have a dialogue stringout, you are left to pull up each shot, one at a time, and navigate to that line and play it. Then, do the same for the next take. By the time you play each further take, it might be hard to remember the last take. For a true comparison of performance, take to take, nothing beats a dialogue stringout.

By placing all the takes together, you get the ability to just pull up the stringout and play them all in a row. Since they are all already trimmed, it's relatively easy to scoot them around and pair the client's favorite two takes against each other to play back to back or on a loop. This allows the direct comparison that is often needed to make the best decision and be confident in that decision.

Of course, you might have noticed that many movies don't cut on every line. Even on projects that favor long takes and letting the shots breathe, it is still sometimes useful to have dialogue string outs. There will often be a specific moment that a director particularly cares about, and the entire take might be chosen off of that one line of dialogue. The director has every right to want to compare against the other takes of just that line. Often the process of settling on "which take" of a master shot will be used is a lot process of looking at string outs for various lines. Only once it's clear that a particular take has a lot of the director's favorite performances is it chosen.

## Media Composer for dialogue string outs

Media Composer has a utility called the script tool that makes the creation of string outs much easier. You import your screenplay, then you are able to tie specific lines of dialogue to specific points in a clip. Media Composer can then create string outs for each individual line based on your syncing process, which is faster and easier than doing it by hand. This is one of a few ways Media Composer continues to really serve the needs of narrative editors, and a feature that is unsurprisingly un-copied by any other major NLE.

As an option, you can even upgrade your Media Composer license to get the "PhraseFind" and "ScriptSync" tools which use voice recognition to provide similar functionality with even less work on the assistant editor. Phrase Find can be used to find every time someone says a certain phrase (such as a line from the script, or a key topic in a documentary), and play them all together. Script Sync builds on the script tool by automatically linking the script to the media using voice recognition. It's not perfect, but it is a valuable tool that many appreciate, especially on projects that assemble a lot of footage.

## Documentary Stringouts

In documentary and reality production, stringouts serve a similar purpose, helping teams watch and evaluate groups of footage. However, in documentary and reality production, it's rare to do a stringout for the entirety of the media collected, since it is often just too much for any production to watch in its entirety. This is a shame, and if the budget allows, everything should be watched, but often budget constraints don't allow this. This is where the field producer's notes come in handy.

On many productions, a field producer is actively taking notes during production of what material might be interesting in

post-production later. If you are shooting a long day of activity for a documentary on auto racing, for instance, the field producer would note times when races happen, times when there is drama in the pit stops, and other notable events that happen through the day.

The editorial crew will then use those field producer's notes to assemble stringouts of likely scenes. For certain productions, this means that some of the footage shot will never be watched at all, if the field producer didn't note that anything interesting happened.

Ideally, someone should watch every moment of video captured for the project. There are numerous examples of magic discovered in an outtake or a long tail that no one would normally watch. In television work, however, that just doesn't happen, and the stringout, driven by a paper edit of notes from the field, is the primary method for sharing footage with the team.

## VFX Stringouts

VFX string outs are a very common task for an assistant to assemble shots to hand over to VFX. While many users who have been working solo for a long time are used to simply dynamically linking to After Effects, bigger shows traditionally use VFX houses that might be all over the globe. In order to properly hand over shots to VFX, it's common to create a stringout.

Every VFX house will have different specifications, but one common workflow involves including both the shot before and the shot after the VFX shot. This is to help give the VFX artist context for their work and better integrate it into a sequence.

The entire scene will also usually be included for reference, but at a lower resolution. VFX artists will want the highest resolution available, typically in DPX format. Depending on the artist or house, they also might want handles on their shots to help with planning momentum or to give the editors leeway in future edits.

## Alternates

Often, towards the end of a project, the entirety of an edit is handed over to another department, such as handing over from editorial to sound and color, or during the final archiving steps of a film.

During this process, it's not uncommon that there are alternate takes, often shortened to "alts," that you want to either hand on to color or to archive. One common situation is including alts for a particular vendor, such as clean no-cursing takes for airlines and broadcasters, or alternate coverage for a specific outlet. For instance, BET has very specific rules on framing, which typically leads most hip-hop videos to produce both "web" and "BET" versions. This is often done by mastering the web version, with the shots needed for the BET version as a "stringout" at the end of the timeline.

When creating this kind of "alt" stringout, it's important to clearly slate shots, since you can't guarantee that future editors who are brought on the project will necessarily easily be able to find where a shot should go. A typical alt stringout will have a slate for each shot the might read "Alternate for Closeup on X at Timecode Y," to help future assistants place the media properly when it's needed.

Many editors will also string out at the end of the film any

shots that received special treatment, such as custom plugins or even speed changes, so they have the un-manipulated shot available to them if they need it later. It is not uncommon in music video and commercial finishing, which rely heavily on effects and alts, to have a stringout with a similar runtime or even multiple times the length of the original spot, though that doesn't show up as much in television and feature work.

Contributed by Charles Haine, Filmmaker, writer, host and professor.

## Collaborative Editing

While all of the popular NLEs have some sort of workflow that allows editors to work together on a single project, none of them actually allows multiple editors to make changes to the same timeline simultaneously (as Google Docs does for word documents, for instance). That scenario would probably not work very well anyway, since an edit to the beginning of the film would "ripple" down, causing issues to anyone working on later parts of the film.

Instead, a distributed editorial workflow requires different editors to "check out" or "lock" a sequence or set of files. When that section of the project is locked, the other editors may be able to view it, but are unable to edit it. When that sequence or folder is unlocked again, it's available for someone else to lock.

So, it's possible to have multiple people working on the same project at the same time, but not *the same part* of that project. Each NLE handles this process differently, some more smoothly than others, but there is a workflow for each of them.

A collaborative editing workflow in Avid

Media Composer is the core of Avid's ecosystem and the basis of many collaborative editing environments. Avid recently restructured the Media Composer line to provide a clear separation of capabilities:

- Media Composer | First – for students
- Media Composer (Standard) – for independent professionals
- Media Composer | Ultimate – for teams of professionals

Let's see how collaborative editing is achieved with Media Composer and Media Composer | Ultimate, which we'll refer to as *Standard* and *Ultimate*.

## Prerequisites

- Install the same version of Media Composer on all team members' workstations.
- Establish a naming and organization structure for clips and bins.
- Provide access to the same Avid project and media.

## The Same Version of Media Composer

Ideally, everyone on your team will use the same version of Media Composer for the duration of the project.

It is possible to hand off projects among editors with different versions, but there may be trouble down the road if you do.

## Establish Your Bin and Clip Naming and Organization Structure

It can't be overstated: organization is the foundation of success in the edit. Before any actual editing begins, establish as a team how bins and clips will be named and organized for the duration of your project.

If the structure can improve, write down your changes, then tactfully present them to the decision maker.

If you're the decision maker, don't get stuck in analysis paralysis, or be unwilling to change. Always be open to the possibility something could be clearer. Listen, then decide. If changes are made, document it, relay them, then stick to it.

When everyone does their part to be consistent with clips and bins, you'll spend less time looking for stuff, and more time making creative decisions.

More details are available here:

Bin Organization

File Naming and Organization

## Access to the Same Avid Project and Media

### With Media Composer (Standard)

As of version 2018.4, Media Composer Standard no longer

natively supports shared storage. So how can you share media without shared storage? This approach works best when handing off the project between individual team members:

1. Ingest footage
2. Log clips
3. Hand off to the Lead Editor
4. Hand off to other team members

Someone on the team, perhaps the Assistant Editor, will create a new project in Media Composer with the desired frame size and frame rate for final delivery.

Getting this right is important. Getting it wrong will mean lost time and lots of work reversing damage later. Verify project settings before creation.

Next, ingest and transcode your source footage to an Avid codec. Your project settings will narrow down which codecs are available for transcoding.

Instead of transcoding, what about just linking to sources so you can hit the ground running? You can, but Media Composer doesn't run at peak performance with most acquisition (i.e. shooting) codecs. On the other hand, Avid's DNxHD and DNxHR codecs do.

Transcode some test shots with the available codecs in your project. Remember, if transcoding to DNxHD, the higher the number at the end of the codec name, the larger the resulting MXF. Try shots with large quantities of random motion, like moving water or a Steadicam-like follow of someone running. How does it look? How big is the resulting MXF in *../Avid MediaFiles/MXF/* on your media drive? Weigh those out, then choose your codec for transcoding.

Media Composer *can* transcode footage in the background while it's running, but reliability varies from version to version. For consistent results, budget time to perform a traditional transcode overnight, or schedule other tasks if transcoding during the day, since you can't Media Composer while transcoding footage.

Name your clips and subclips according to the agreed-upon organization structure. Adhering to that structure is crucial. Naming clips in Media Composer is a manual process, and there's no batch renaming tool in Media Composer.

Multicam clips may be built and additional metadata, such as script notes, may be imported or added as well.

At this stage, the project's ready for the Lead Editor. The initial handoff consists of gathering and transferring the Avid project folder and all associated media.

Media Composer follows a consistent pattern for storing projects across Windows and macOS. Remember that project you created? More than likely, you'll see a folder with that project name inside your home folder at this location:

- Windows

° Private Projects: *C:\Users\\Documents\Avid Projects*

° Shared Projects: *C:\Users\\Documents\Avid Projects*

- macOS

° Private Projects: */Macintosh HD/Users//Documents/Avid Projects*

° Shared Projects: */Macintosh HD/Users/Shared/Avid Media*

*Composer/Avid Projects*

Copy that folder to an external drive or a shared folder, which will be transferred to the corresponding folder on the Lead Editor's workstation.

You *can* use Media Composer's built-in Media Tool. But doing so is challenging for all but the most seasoned Media Composer user. Instead use MDV to gather project media.

MDV (Media Database Viewer) is a free media management tool written by a veteran Media Composer user, available for Windows and macOS. It scans all Avid MediaFiles folders on all mounted volumes, then displays all available media files associated with a project. MDV can also scan individual bins, showing all available media connected to clips in that bin. Using it, you can sift media for an entire project or individual bins with clarity and confidence. It'll even sift the media into new folders named after the project in question.

Sidebar: all media imported or transcoded into Media Composer is managed by Media Composer. The media are usually MXF files inside a special folders on your designated media drive. Those folders are under *../Avid MediaFiles/MXF/* (where N is a number of 1 or higher) and contain databases that index and connect the MXF's to the clips in bins accessible from Media Composer.

Using MDV, copy the project media to a folder on an external drive or a shared folder. These MXF's will be transferred to the Lead Editor's media drive. The simplest approach? Quit Media Composer, then create a newly number folder inside /Avid MediaFiles/MXF. If you already have a "1" folder, name the new folder "2", then copy the MXF's there. Relaunch Media Composer then go for a walk while it indexes the new media. Open the project on the Lead Editor's machine, and they're ready to go!

More details are available here: http://djfio.ru/mdv/

What if an Assistant Editor needs to add more to the Lead Editor's project? The Assistant can add new media by creating new bins in their local project where they'll ingest, transcode, and log. When they're done, they'll transfer the new bin along with all media gathered using MDV to the Lead Editor's workstation and media drive.

## With Media Composer | Ultimate

Avid introduced Media Composer | Ultimate in version 2018.4. Unlike *Standard*, *Ultimate* can take advantage of the collaborative workflow feature set available in a traditional Media Composer environment, such as connecting to shared storage with Avid file systems (AvidFS) or bin locking in shared projects. *Ultimate* can also connect to MediaCentral, Avid's media asset management platform, for extended functionality.

More details are available here:

Bin Locking and Simultaneous Access

## Beyond Media Composer

At NAB 2018, Avid announced *Avid On Demand*. *On Demand* provides access to editing and media management tools in a Software as a Service (SaaS) model with cloud-based storage supplanting large shared storage appliances. Instead of deploying, administering, and supporting costly facilities, Media Composer seats can be added with applications 'spun up' or 'spun down' (like a mechanical hard drive) as needed. Or, on demand.

More details are available here:
https://www.avid.com/solutions/cloud

### Handing off to Other Team Members

After picture lock and conform, you may be working with a colorist and an audio post specialist during the finishing phase. The handoff typically consists of an AAF file along with a reference export of your movie and media sifted from your project. Media Composer can export AAF's tailor made for round-tripping in DaVinci Resolve and Pro Tools, which you can save as reusable export settings later.

## A collaborative editing workflow in Premiere

### Overview

For a long time, "shared user workflow" in post meant working with Media Composer or Resolve. For many post workflows, it's incredibly useful to have more than one person working on a shared set of media or timelines (editor and assistant editor, for instance, or story producer and director, or any other number of combinations). But Adobe Premiere and Final Cut Pro focused mostly on making life as easy as possible for the "one mule team" filmmaker, developed more and more functionality for those who could work off of a single system and neglected multi-user features. After the demise of Final Cut 7, Premiere took off with a growing user base in the social video and marketing space, but couldn't really break into television and feature film use without better multi-user toolsets.

For situations where you absolutely had to have multiple users on a project, there were frustrating workarounds. One old method was simple but somewhat painful: share .prproj files back and forth between users, while keeping identical media on multiple separate machines. This required tremendous work as projects grew complicated, since every time new media would come in, you'd need to make sure that the editor, assistant, director, and more were getting copies of the assets. Keeping track of the "current" version of the project was a tremendous hassle for the assistant or post supervisor, and if two people worked truly at the same time on the same timelines, there was no way to reconcile their work together.

On a shared network, more than one user can of course open up a premiere pro project file at the same time, but whoever clicks "save" last will overwrite all the work done by another editor while they are working, which is hardly an ideal solution and requires tremendous communication from room to room in the facility to ensure that nobody accidentally clicks save and destroys another editor's hours of work.

The last few years have seen Adobe take multi-user workflows more seriously as a method of moving in on Hollywood feature and television work, enabling workflows that will also have an appeal to those working in smaller facilities where it's still sometimes a real timesaver to have even two users on the same project at the same time.

For maximum ease of use, this requires all the users to be pointing at the same pool of media. Right now, the best functionality for multi-user situations is to work on a shared network, though there are some options coming down the road for cloud sharing that we discuss below.

## Shared Network

Unlike a platform like Media Composer, which still works best when all media is ingested into a dedicated media folder, Premiere allows users to leave their media wherever they like. A user could potentially edit directly off of the camera capture card if they chose, though that isn't always the best workflow if the card is soon to be reformatted. In reality, this means that users will frequently keep their media on a variety of external drives and occasionally on their desktop, which doesn't scale well multiple users. Media Composer will actively migrate all your media to a media file on a server if you set that as your media drive, and Premiere can do the same if you are smart with your ingest settings.

For a multi-user workflow, you will want all of your media to be stored in a central server of some sort, generally referred to as a SAN for "storage area network." With hardware solutions from LumaForge, EditShare, Drobo, and even Avid whose Nexis servers can work well with Premiere, there are many options to choose from, but the key is having a server that is specifically designed for multiple users to be pulling from the same set of files at the same time.

We recommend creating an individual folder on your server for each project you work on and having a "media" folder within that. That will make it easier to archive a project after it's delivered; rather than going into the master media folder and trying to figure out what goes with what project, you move the entire master folder, with all the media, to your archiving solution all at once and it takes all the media with it.

Premiere generally links to media, meaning it leaves it where it is, but you can set up Premiere to automatically ingest media (moving it to a new location). When setting up several editors on shared storage, it's often worth doing that to make sure that all the media ends up migrating to the shared storage and nothing lives on a local editor's internal drive, where file path routing can lead to lost media. Nothing is more frustrating than having one of the editors save dozens of

assets to their personal desktop or external drive, making them unavailable on a day the editor isn't around. Setting up for ingest, instead of linking, avoids that situation.

You can set this ingest path on the launch window for new projects, and if you are going to be using a lot of multi-user workflows, you might want to customize the default settings on all the machines on your system.

## Multiple Projects at Once

Central to understanding how Premiere works for collaborative editing is the idea that Premiere allows you to have several projects open at the same time. While most editing applications open one project at a time (though Resolve now allows for keeping more projects "active" with fast project switching, it's memory intensive and you are really only keeping one project actively open at once), Premiere allows you to keep several projects open all together. When you open more than one project, you'll see tabs at the bottom of your project window for your open projects, and you can close them one at a time.

The benefits of this are massive. Whereas previously, if working on a 6-episode series, a user might be tempted to put all 6 episodes together in one project. Putting all episodes in one project makes it easier to switch back and forth between episodes to avoid repetition and to create smooth transitions. With multiple projects open it's now possible to split episodes into their own projects. This saves system memory, as you might only need 1-2 episodes open at a time instead of all six, while allowing users to easily move sequences and media between projects without having to wait for a long process of closing one project and opening another. Simply copy out of one project, switch over to the other project, and paste.

## Project Locking

With no projects open in Premiere, if you navigate to your preferences panel for collaborative workflow, you can enable "project locking." When you turn on "project locking," only one user can make changes in a project at a time. If other editors open the project, they will only have "locked" access, where they can see all parts of the project but make no changes. They can tell they are restricted from working by the small "lock" icon in the lower left of the projects panel, which will be locked and red, when working with a project that is locked.

This is tremendously useful in scenarios when many people are working together to try and get a project over the finish line. Let's say you are working on episode 5 and want to see the end of the current edit of episode 4, even though an editor down the hall is currently working on it. You can easily navigate to the project file (provided it's saved on shared storage and not on a local hard drive, of course), and open it in locked mode in order to review what it looks like currently without interrupting the other user's work or having to ask them to close the project.

You can also leave projects locked even if you are the only user editing that project at that moment in time. For instance, if you know you aren't going to be making changes and want to leave the project available for other users. Or if you know you don't want to make any changes and want to avoid accidentally damaging an edit in a way you might not notice. Just click the clock icon to lock it up, and the next user to open it will have it available unlocked by default.

Projects only refresh on opening: if a user is working in another workstation, the changes won't dynamically ripple to

your workstation. You'll need to close and re-open to see the changes they have saved.

Multiple open projects will eat more of your system resources, especially memory, so be sure to close extra open projects using the file menu or the hamburger menu next to the tab on the bottom of the projects panel.

## Team Projects and Premiere Rush

Adobe has been pushing their "Creative Cloud" branding for a while now, with subscription pricing and cloud storage included in a bundle. It should be no surprise that they are at the forefront of pushing collaborative technologies that are built not just around the local area network, but around keeping your media on the cloud to be accessed remotely by teams separated by distance.

At the top end of their offering is Team Projects, which allows for multiple team members to access the same pool of media simultaneously on the creative cloud servers. Only available for enterprise and business customers, it works by creating low-weight H.264 proxy files on the web that can be streamed to editing stations in various places when necessary.

A new product that has just rolled out but shows some collaborative promise is Premiere Rush. Originally launched in beta as Project Rush, Premiere Rush is a platform designed to move your media seamlessly in between workstations. By creating low res H2.64 proxies, you should be able to seamlessly open up a Premiere Rush project on your desktop or your mobile device and have it all point right to that same media on the creative cloud server.

In one scenario, you shoot on set and download with a

laptop. That laptop makes background H.264 proxies and uploads them to the Premiere Rush server, so you can work on the Premiere Rush iPad app while taking the train back to the office, where you can then fire up Premiere Rush on the office iMac to make final revisions.

Premiere Rush doesn't currently allow more than one user to work on the same project at a time, but it does create a synced pool of media on the web that can be accessed by multiple machines. Two editors who are collaborating on a project (or a director and an editor) could share a login and take turns working on the same timeline. It requires some communication to make sure they both aren't on the same project at the same time, but it appears possible.

In order to offer a web-friendly interface that works on iPad and iPhone, Rush currently has a reduced feature set, but its projects can be ported from Rush to full-fledged Premiere. While its potential for teams is still being explored, it seems likely we will see situations where assistant editors start the process of creating a stringout, highlights reel, or even a rough cut from set in Rush before handing that over to a lead editor to do final work in full-fledged Premiere, all pulling from the same set of web media seamlessly.

Contributed by Charles Haine, Filmmaker, writer, host and professor.

A collaborative editing workflow in FCP X

## A collaborative editing workflow in FCP X

Collaborative editing *can* be done in Final Cut Pro X. The

approach is based on these principles:

1. Between Editors
2. Between Editors and Assistant Editors
3. For Finishing

Note: we'll use Final Cut Pro X parlance from here on. But for clarity's sake, here's a legend of essential FCP X terms:

- Library – the structure containing your timelines, media, and clips. (Essentially the "project file" for FCP X.
- Project – a timeline or sequence. (Not to be confused with the traditional meaning of a "project file.")
- Event – the container in a Library where clips are organized by keywords.
- Clips – references to underlying media in your Library, whether managed or external. Available in the Event Browser.

## Establish Your File Naming and Folder Organization Structure

It can't be overstated: organization is the foundation of success in the edit. Before any actual editing begins, establish as a team how files and folders will be named and organized for the duration of your project.

If the structure can improve, write down your changes, then tactfully present them to the decision maker.

If you're the decision maker, don't get stuck in analysis paralysis, or be unwilling to change. Always be open to something could be clearer, listen, then decide. If changes

are made, document it, relay changes, then stick to it.

When everyone does their part to be consistent with files and folders, you'll spend less time looking for stuff, and more time making creative decisions.

## Decide Where Library Media Will be Stored

### Shared Storage or Local Storage?

Working with multiple users on the same network in a facility? Chances are you'll use their shared storage for media access. Here's how it works: a facility's IT staff will grant you access to a portion of the server. Then they'll show you how to mount and access any files and folders available to you.

If the facility doesn't use shared storage, or when working remotely, use the highest-performing local storage available to your computer. On this drive:

- Recommendation: do not use your startup disk as a media drive.

### Import Media Inside or Outside Your Library?

A Library is a *.fcpbundle*. The *.fcpbundle* is a group of files and folders that represent all the Projects, Events, and Clips you see in FCP X. Some of these folders are designated for media. Depending on how you import media, your Library can quickly balloon with a single import.

Knowing that, there are two ways to import media:

1. Managed Media – imported media is copied into (read:

inside) the Library. In this way, media locations are managed by FCP X.

2. External Media – imported media isn't copied inside the Library. Clips are linked to media that live in shared or local storage. So media locations are external to FCP X.

When working with a team, import your assets as external media. If everyone's following the file and folder organization, it ensures you're viewing and accessing the same media as the rest of the team. It also keeps your .fcpbundle small for faster handoffs.

## The Handoff

Currently, handing off a Library is the core of collaborative editing in FCP X. How that happens, natively and with third-party solutions, highlights FCP X's flexibility when working in teams.

### Handing Off Libraries or XML's?

Even if you're a band of two, handing off FCP X Libraries, *.fcpbundle* files, is the best way to maintain the integrity of the work among multiple editors.

If the Library lives on shared storage in a facility and you're sick or on vacation, no problem. Your teammate can hop on the network, open up the Library, and the work can continue.

If you're working remotely, make this your habit at the end of the day:

- Check [Delete Render Files].
- Select [All].

If you're sick or on vacation, and (wait for it…) everyone's following the file and folder organization, your teammate can just download the Library with any new assets, relink, and the work can continue.

That said, here are some caveats with Library handoff's:

- A newer version of FCP X will open older Libraries, but an older version of FCP X can't open newer Libraries.

- Effects
- Transitions
- Titles
- Generators

Note: the Motion Templates folder is located in your local macOS account under */Users//Movies/Motion Templates/*.

All team members should have the same fonts installed if any text or copy is used in the edit.

Of course, none of these are insurmountable. Third-party content in Motion Templates can be bundled along with your Library before handing it off, or installed on your teammates' workstation.

You might wonder why FCP X versions might not have to be the same. That's where XML comes in.

FCP X can export Libraries or Projects as *.fcpxml*. Since XML is just text with markup, it can be a lightweight method of sharing Libraries if a native Library is too big to hand off. XML is another way of getting a Library to another editor with a different version of FCP X.

The main caveat with XML handoffs is the potential for loss in translation. Sometimes obscure elements of your work don't make in the export. For example, if you're working with interlaced source footage, and you deinterlaced it in FCP X, the "Deinterlace" flag may not show up if the XML is exported incorrectly. Even when handing it off with notes, the other team member may not notice.

For maximum output fidelity, favor Libraries over XML's for handoffs.

Now let's talk about different handoff scenarios.

### Closed handoff

If you're starting the edit, while another may finish it, the handoff is simple. Store your Library in the agreed-upon folder, or upload it to a cloud storage location. Your teammate can open your Library and continue working.

### Between Editors (Basic)

But what if you're working on one Scene 1, and your teammate's working on Scene 2? Given both of you have access to the same assets, each editor can create their own Libraries with Projects in different Events, which can later merge into a project in a master Library.

- Final Cut Pro X will prompt you to copy new media from those other projects.

### Between Editors (Advanced)

But what if you're both working on the same scene? Or the same episode? Remember, handoffs are the core of

collaborative editing in FCP X. If you're working from shared storage in a facility, you can't open a Library if someone else has it open. Enter these third-party solutions.

Released: September, 2017

Here's what Final Cut Library Opener can do:

- If another editor has a Library open and you need something from it, FCLO creates an ephemeral (or temporary) copy of that open Library.
- Once created, you open the ephemeral copy of the Library and grab the Event(s) with stuff you need.
- When you're done, close the ephemeral copy of that Library and FCLO deletes it automatically. Hence, ephemeral.

Additional notes:

- If it's not in an Event, a quick call or message might take care of that.

More details are available here.

Final Cut Library Opener may be all that's needed to address certain speed bumps to progress. But what if your team needs something more?

Released: September, 2018

PostLab is the most comprehensive solution for collaborative FCP X workflows to date.

With PostLab, teams can:

- Build a versioned repository of Libraries, complete with

comments.

- Open Read-Only copies of Libraries.
- Integrate popular Web services used during postproduction in one UI. Services like media asset management systems, Slack, and Frame.io

Additional notes:

- PostLab is open source. So it's provided and updated at no charge.
- A GitLab server is required. PostLab provides video tutorials on how to set up your own server using GitLab's public servers or building your own.

## Between Editors and Assistant Editors

How about when Assistant Editors are prepping footage or doing assemblies?

Library handoffs and mergers, plus Final Cut Library Opener or PostLab, may come into play. But there's one more tool you might add to the mix.

If an Editor and Assistant Editor are working on their own Libraries with the same Events and Clips, the Editor could be adding keywords to Clips and marking Favorites while the Assistant Editor is doing their own keywording and Favorites.

MergeX takes both of these Libraries exported .fcpxml, and merges metadata for all matching clips.

## When Handing Off XMLs Is Required

Handing off XML's might be necessary if:

1. A teammate's version of FCP X is different from yours.
2. You're using a third-party media asset management (MAM) system. Most FCP X-compatible MAM's only allow you to add an *.fcpxml* of your Library, not an actual Library.

But handing off XMLs is most useful when working with Finishing team members who aren't using FCP X, such as:

• Colorists – DaVinci Resolve

• Audio Postproduction Specialists – Pro Tools, Logic, REAPER

Note: before exporting XML for the handoff, some additional Project prep is required.

If color finishing is done outside of Final Cut Pro X, most likely it will be in DaVinci Resolve.

Resolve and FCP X have enjoyed a favored relationship for a long time. Export your Project as XML (File → Export XML), import that .fcpxml in Resolve, and Resolve is off to the races.

The result: a .fcpxml exported from Resolve, then imported back into FCP X. This .fcpxml is a new Project representing the final delivery with graded and/or conformed footage.

More details are available here.

If an audio post session is also part of the finishing phase, taking the time to designate Roles for all audio sources in your Project that will make the conform exponentially easier.

Logic can import *.fcpxml* directly.

Pro Tools requires an *.aaf*. You can convert *.fcpxml* to *.aaf* using other tools like:

- X2Pro (costs around US$100)
- DaVinci Resolve (Free – here's how)

Vordio is a workflow tool that takes *.fcpxml* and converts it to a REAPER session. It's non-destructive (re)conform process and strong visualizations between session changes make this a powerful option if you or your team are interested in using REAPER to finish audio.

The result: a finished audio mix is exported, then imported into a Library back into FCP X for insertion into a Project for final delivery.

FCP X's focus on organization through metadata can help your team stay focused on keeping assets organized. You can work with Editors, Assistant Editors, or Finishers using FCP X's native toolset, or extend your workflow with a rich set of affordable (or even free) third-party options. Yes, collaborative editing in Final Cut Pro X isn't just possible. It's accessible. And powerful.

A collaborative editing workflow in Resolve

## Overview

Before DaVinci Resolve expanded into audio (Fairlight), VFX (Fusion), or even editing, the software was intended purely for high end color correction. Because they were targeting use at highly-networked color grading facilities, it was always built

around a database model with multiple simultaneous or nearly simultaneous users in mind. This would allow a colorist to be "in session" in one room with a client, while a color assist could prep, do renders, or even do color cleanup in another room on the same project, if they were pulling from the same shared media.

While the setup was originally a bit complicated for the average user, with the release of Resolve 14 in 2017, Blackmagic started to take shared-user workflows more seriously, with a host of new tools that allow for very sophisticated collaborations, and with a toolset to make even the initial setup easy for post professionals who might not necessarily be computer engineers, but who can walk through a bit of IP network setup.

Resolve comes at two price points: the free software, which is actually incredibly powerful, then "studio," which costs $299 and has a few upgrades that generally make the price well worth it. On top of noise correction and some other plugins, one of the big benefits of Studio is that it enables shared user features, and will be required for setting up a shared resolve environment.

The whole concept of sharing in Resolve is built around a central database, which is something that sometimes confuses new users to resolve since all your resolve projects are stored in a database of some form.

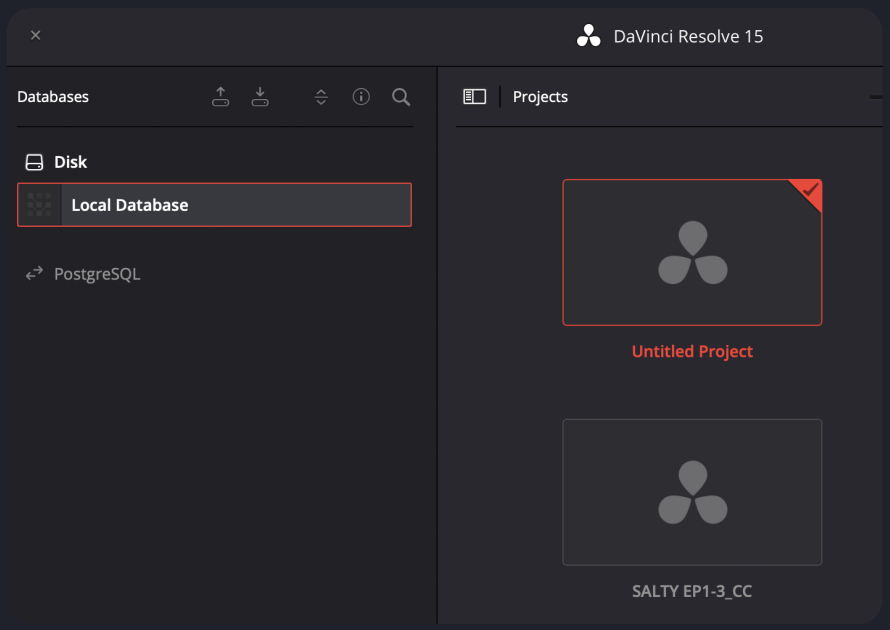
## Working with a Database

Most post production applications work with a project file that is independent of the program and can be moved from machine to machine relatively easily. With the .avp and .avb formats from Media Composer, or .fcp projects from Final Cut 7 and Premiere Pros .prproj, the concept of a "project" is

something most posties have a handle on.

Resolve does have a project file, the .drp (Da Vinci Resolve Project), but it's only used as a method of backing up your work and moving it to a new machine. If you double click a .prproj file, it'll launch premiere and keep working from and saving to that project file. If you double click a .drp, it will launch Resolve and load that project, but it loads it into your Resolves database and works from and saves it there. It doesn't keep updating that .drp file.

Resolve keeps all of its files in a database to make it easier to share. If you set up a shared database correctly, all users on the system can log in to the database and open the same project at the same time. Since the database server is a sophisticated tool that allows for reconciling multiple versions of the project in a powerful way, this allows for a number of powerful collaboration tools that don't show up in other systems that rely on read-only modes to enable opening the same file at the same time.




Resolve can actually deal with two sorts of databases:a disk database stored on the user's local machine and a SQL database, which can either be "served" from your local

machine or from a machine on the network you are attached to. The default behavior is a disk database, and that's how the vast majority of users who work with Resolve and learned it in the last few years are used to working. But if you open up the tab to the left of the projects in your database, you'll see it's possible to attach to a "PostgreSQL" database, which is necessary for a shared workflow.

## Setup

Blackmagic have actually made it easier than before to create a shared database with the new "Da Vinci Resolve Project Server" application. Whereas before it required entering the terminal, which can potentially make some users nervous, with the app you can set up a shared database quite quickly.

However, before launching the database, you need to make sure that your hardware is se tup correctly. First, you need a machine dedicated to "serving" the database, ideally a machine that can be always on on your network that will run PostgreSQL and share the database with other users on the network that need it.

On top of that, you need to be able to assign a fixed IP address to that machine, so that other computers can find it. Generally, most smaller networks assign IP addresses dynamically: every time a new machine logs into the network, via ethernet or wifi, an IP address is randomly assigned to it. For shared databases to work, you want to assign a static IP address to the machine serving the database, and ideally to the machines that will be looking for it. This allows the connection to stay in place even if a machine or the network is restarted. While assigning static IPs is a bit beyond this guide as it's different for every router type, it should be easy to find information on how to do it on your network.

Once you've fixed the IP addresses, you can launch Da Vinci Resolve Project Server, which comes with Resolve Studio. This application has only one function: creating a shared user database. You'll want to launch it on the machine you are using to "share" your database, but that machine doesn't need to continue running Resolve or in fact keep any application "open." The database server software runs in the background.

From that application, you can both create a new SQL database, you can also enable sharing and, importantly, create a database password. This is a powerful tool, since many facilities need to worry extensively about who has access to what. Let's say you are working on promo material for both the new DC movie and the new Marvel movie, as you might in a trailer or advertising company. Both clients will want to be sure that only the creatives assigned to the team have access to the projects. Thus you might create a dedicated "DC" server and a dedicated "Marvel" server to keep the projects completely separate, and only distribute passwords to the right team members.

Back at your workstation, you then would click "Add Database" under the database list, and log in to the database server using the IP address of the database server hardware. This is why static IP addresses are vital, since your workstation will be using the IP address to look for the shared database.

## Shared Storage

Like most shared workflow solutions, Resolve works best when working off of a shared media storage solution, such as a server. That way, every single user is looking for media at the same place.

Many users, when setting up their shared resolve database,

are initially inclined to focus on setting up the fastest possible shared database server, but in reality sharing the database is nowhere near as intense as sharing the media between multiple machines. In fact, it is not uncommon to see an older server, or even a Mac Mini, set up to serve the database, since, while some CPU power is important, as is maxed-out RAM, the server doesn't really need to be a powerhouse, especially in GPU power.

Your shared storage, however, should be ready to play out media with enough speed to handle as many users as will be working on the project, and that is where some investment is worth it. Some systems, such as the Jellyfish by LumaForge, even come with a shared Resolve server built in. However, if your shared media server isn't ready to play easily with a shared Resolve database, a used Mac Mini will do just fine.

## Read-Only Mode and Locking

Once you have your media on a shared media server, and two or more workstations logged into the shared database server, it's time to start experiencing the power of multiple users on the same project at once. One of the important concepts to understand that enables the system to function is "read only" or "locked" versus available resources. While Resolve doesn't "lock" entire projects as other applications might, it does "lock" certain sections within a project when it's being used by a given user.

To turn on project sharing, you right click on the project in the projects window and enable project sharing. That allows multiple people to log in to the same project at the same time. If another user logs in to the same database, they should see the project there with a little symbol indicating it is available for multi user work.

## Editing and Timeline Compare

It is within the project that this becomes more complicated. Resolve doesn't let two people actively edit the same timeline at the same time. When the first user logs in, they take control of the timeline, and other users can only view that timeline in a read-only mode that allows full viewing, but no changes.

For most workflows, this is a very useable solution. For instance, a lead editor can be busy working away on a timeline, and the assistant can come in check on progress, or to confirm for VFX that yes, actually, a certain shot has been cut. You can copy from that locked timeline into other timelines, or most powerfully, you can also create a new version of that timeline to work with. At the end of a session, the owner of the timeline can then go through an "integration" step to compare changes from both timelines and "commit" the changes from the second user to the master timeline.

Resolve has built a powerful view that shows the changes in the timelines graphically, as well as a list view, making it easy and fast to commit or reject changes that a second editor might have been experimenting with in the timeline.

## Multiple Colorists on the Timeline at once

While navigating multiple editing timelines is relatively easy, navigating and then merging multiple color sessions would be much more complicated. For instance, when color grading a shot, there are literally hundreds of things you can do to manipulate it; creating a merge session window if two colorists were working on the same timeline would create a list thousands of items long by the end of a day.

Resolve solve this problem by not color locking at the timeline level but at the shot level. When working in a multi user project, when the colorist is working on a shot, only that individual shot is locked, but the shots around it are not. Another user won't be able to work on the same shot at the same time, but the shot before or the shot after is of course fair game.

This is a powerful tool but also a dangerous one. Of course, colorists are in the habit of doing a final watch down (or three) to ensure that all grades are what they want them to be, but this multi user workflow does allow for a colorist to be working on shot 10 while another colorist makes tweaks to shot 3 that the lead colorist might not notice. Shared workflow coloring requires a highly trained team who are used to working together and have developed clear protocols for who is doing what within a project.

## Realtime Chat

One of the nicest features that Resolve has added to help users communicate who is doing what is a realtime chat window built within the interface. When multiple users are logged in, the chat icon appears in the lower right, and allows for chatting to occur with any other user currently in the project.

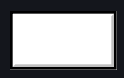
While, yes, many chat applications exist, if you are in a room with a client, you never want to leave your primary software. Having the ability to chat within Resolve, rather than switch over to gmail and explain "oh, we use gchat, I'm not checking my personal email" is a huge benefit to users.

This allows a colorists to ping an assistant "hey, I'm having trouble tracking shot 8 but want to move on with client, can

you clean up the roto by hand?" and keep moving on creative work with a client without worrying about drawing the absolutely perfect shape. Or an assistant editor can ping the lead editor "hey, done with the stringouts on that line if you want to look at options." By default, it comes up with your computer name, but you can of course change it to whatever else you might like.

Contributed by Charles Haine, Filmmaker, writer, host and professor.

## Stages of the editing process



The number of stages and amount of time spent will also vary widely based on the type of film. A short film's editorial process may only take a week or two, while a feature film may take 5 or 6 months to edit. On a documentary film, the editorial process often runs in parallel with production, since the story is built primarily in the edit. Some documentaries have literally taken years to edit as the filmmakers patiently and meticulously gather interviews when time (and budget) permit.

Many feature films, even blockbusters, will begin their post-production during the shoot as well, with the lead editor on-set putting clips together, or back at base, cutting completed scenes.

"Mission: Impossible–Fallout" lead editor Eddie Hamilton began putting together scenes for director Chris McQuarrie during the production phase.

The process is still fairly standardized among different types of projects, though some phases will be much condensed.

Stages of the Editing Process

## Assembly Edit

While the production team is shooting the film, the editorial team is often just a day or two behind them, building the assembly edit slowly and steadily from the previous day's dailies. The editor has to move quickly and only has access to part of the film (the part that has been shot so far). So the assembly will be very rough, but it can provide the structure for each scene and ensure that each scene works as shot.

It can be very helpful to the director to have the editorial team working in lockstep because the editor will be able to spot any potential issues very quickly. Perhaps a mistake was made and

an important shot isn't usable. If the editor can spot that within a day or two, it may be possible to do some reshoots before the production team has moved on to another location.

On a short commercial or music video shoot, the entire production may only last for a few days, with no room in the budget for reshoots, and so the turnaround time for the assembly may need to be even quicker.

The assembly edit is usually very close to the script and includes every line and piece of action that was shot. The assembly thus is usually quite long, sometimes more than twice the length of the final film. There are no visual effects, no color correction, no sound effects and usually not even temp music.

While the structure of the story is all there, the assembly is usually pretty far from the finished product.

## Rough Cut

Most of the blood, sweat, and tears of the editing process are shed in getting the film from the assembly to the rough cut, which may take several months of hard work.

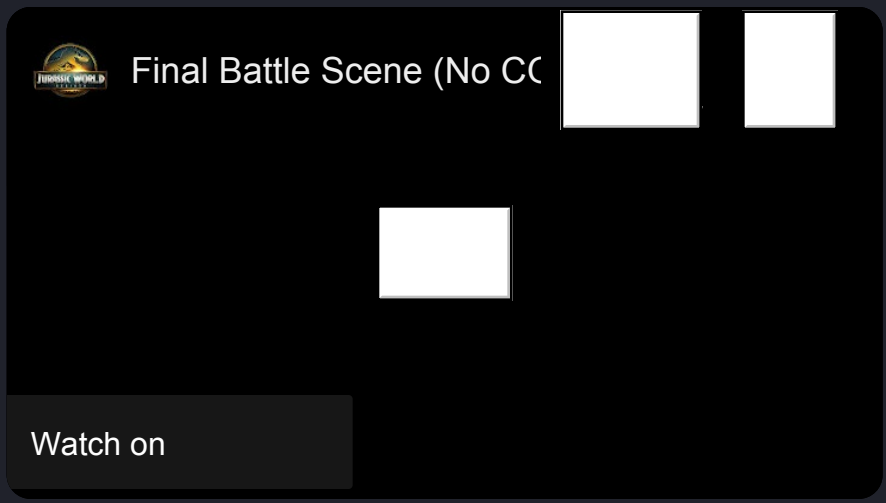
In most cases, the director works closely with the editor during this stage. Because the director was required on set throughout production, the editor has usually been working without a lot of input in building the assembly edit.

For some scenes, the process may be as simple as taking the assembly edit and trimming it down and tweaking. In other cases, the editor and director may start from scratch with a completely different approach.

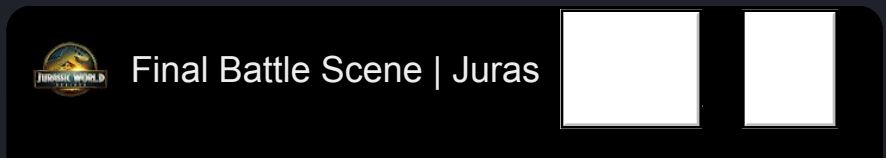
If the film has VFX in it, the rough cut probably will only show very basic simplified versions of the VFX shots. No professional sound people have touched the film yet, so the sound effects and sound mixing will have rough edges.

Once the rough cut is done, the film is usually shown to the client or producers for feedback, but that sometimes depends on who those people are. The producers of a Hollywood film are used to seeing rough cuts and are usually able to see past the many flaws of this version of the film to judge how the story is working. The client for a marketing video, on the other hand, may not have the experience to judge it properly, and so it may be a good idea to wait for a more polished version before getting feedback.

For example, the Jurassic World team released a sequence on YouTube that's edited just with the dailies (no final VFX included). It's quite strange to watch, since the dinosaurs play such a huge role in the sequence. It takes a great deal of imagination to be able to judge whether the sequence is working, at this point.




You can view the finished sequence here:



If reshoots are needed, it should hopefully be clear at this point, so that they can be scheduled in time.

## Intermediate stages

Depending on the type of project, there may be several more milestones after the rough cut. There may be a "first cut", which is shown to test audiences, especially if there is a concern from the producers or disagreement between the director and producers about how audiences will react.

There may also be a "fine cut," which is shown to producers and others involved in the film's production if the editor and director aren't comfortable showing the rough cut.

On many feature films, the director will finish working with the editor, creating a "director's cut," and then leave the film to work on their next project. The producers, however, may request more changes to be made, resulting in a "producer's cut" which ends up being the final version. Unless the director is also a producer and/or has a role in the financing of the film, he or she seldom has the last word.

## Picture Lock

After much work, many versions, many arguments, and many

compromises, the picture is "locked", which means that no more editorial changes can be made. No cuts can be altered. The film is far from finished at this point, but editing decisions cannot be changed.

It's essential to reach this stage and to actually *lock* the picture lock so that the color, sound, and finishing teams can begin work. If changes are made to the edit after the film has been handed off for the next stage of postproduction, that can cause all sorts of issues (and extra costs).

In an ideal world, the VFX is also delayed until picture lock, at which point the shots that need work are handed off to the VFX houses. It's wasteful and demoralizing for a VFX house to spend weeks perfecting a beautiful animation only to have the editorial team decide that, in the end, they don't actually need that shot.

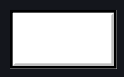
In the real world, however, there is seldom time to wait. If the film is an action or sci-fi film with many hundreds of VFX shots, it's often not realistic to wait many months between picture lock and the film's release. Or if the VFX includes complex CGI (an entirely digital character, for instance), the VFX houses will need to begin work long before picture lock.

In those scenarios, the editorial team may focus on finishing the most VFX-heavy sequences before the others, so that at least those individual sequences can be locked. The unfortunate fact is that, sometimes, VFX work is done that cannot be used.

If you ever watch deleted scenes from a film, you may notice shots with lots of green screen or where digital characters are missing entirely. That is a very good thing – it means that the scene was deleted from the film *before* time was spent crafting the VFX.

Contributed by Reuben Evans, Faithlife TV.

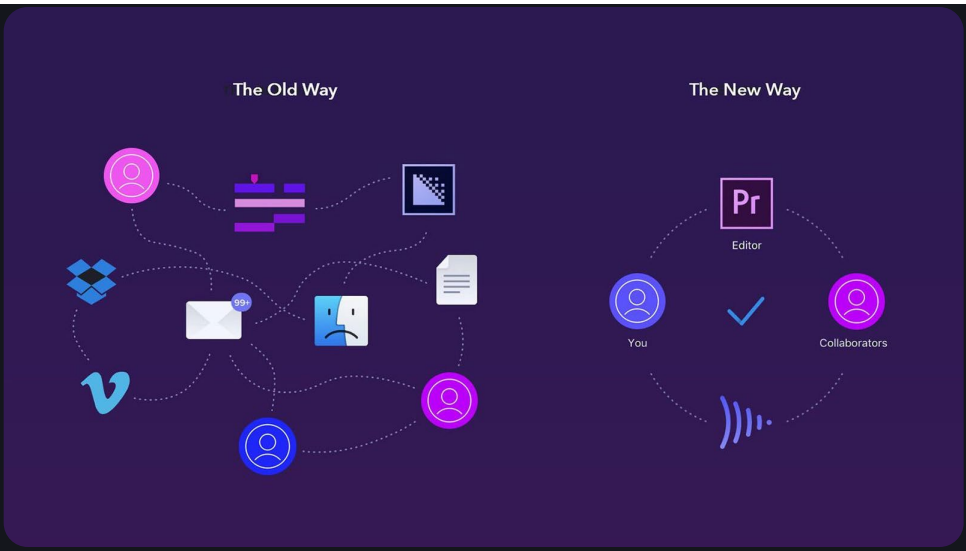
## Review and feedback



For decades, the only way for the editor to receive feedback was for the director or producers to come directly into the edit suite, and that is still common on feature films and broadcast TV workflows. Many edit and color correction suites include couches, kitchens, and large TVs which allow the client or producer to view a work-in-progress comfortably.

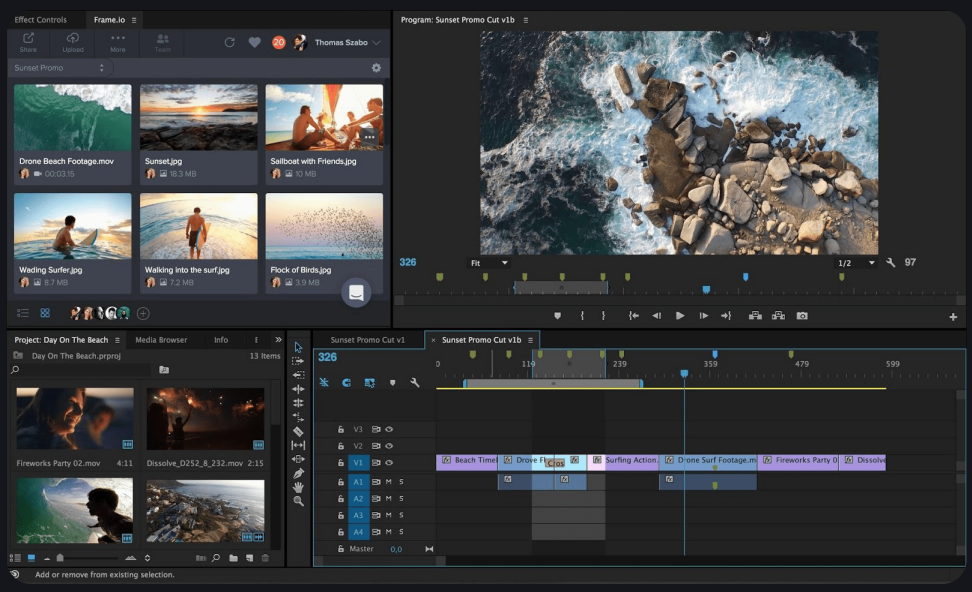
The greatest advantage of this setup is that it allows the reviewer to interact directly with the editor as he or she works, giving comments and suggestions that the editor may be able to implement on the spot. That setup can also be a downside, however, as many editors find it frustrating to work with a reviewer literally standing over their shoulder, especially if the reviewer does not have a deep understanding of the editing process.

Online feedback as an alternative is growing rapidly, as increasing internet speeds have made it easy to upload new drafts of work-in-progress edits. In the past, most video collaboration involved uploading a video to a storage site or video sharing site, then a back-and-forth barrage of emails would ensue among all the collaborators. This process makes it difficult for the editor to manage and process all the feedback, especially if there is a large number of collaborators on the team.

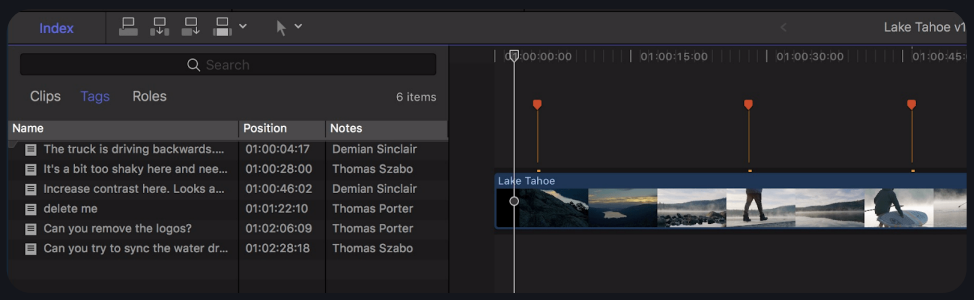



More and more productions of all sizes (from blockbuster films and TV shows to wedding videos) are using services like Frame.io to centralize all their creators, collaborators, and audience. From one platform, video creators can share content and receive organized feedback from collaborators, all in one location.

As soon as a draft is ready for review, the timeline is rendered and compressed, the file is uploaded, and the reviewers are notified automatically. Each reviewer (director, client, producer, etc) reviews the work-in-progress directly from a web browser or smartphone and sends timestamped comments back to the reviewer. Those comments can be downloaded into the NLE as markers so that the editor can respond with new changes.



Frame.io Panel inside Adobe Premiere Pro




Comments from Frame.io can be dropped into FCP X as Compound Clips.

The major advantage of online review and approval tools like Frame.io is that they allow both distributed and asynchronous feedback.

Distributed feedback online

As video production becomes more and more global, the ability to provide feedback from anywhere in the world becomes more valuable. While a traditional review process requires all of the stakeholders to be in the same room at the same time, an online review process can happen from anywhere.

Asynchronous feedback online

With asynchronous feedback, different reviewers can leave their comments at different times, which is often even more important than distributed feedback. Many teams use Frame.io for review and feedback even though the editors and reviewers are in the same building. It would be easy to walk into the edit suite and give verbal comments, but asynchronicity allows each reviewer to give feedback when it's most convenient for them, instead of having to schedule a meeting.

Another benefit of asynchronous feedback is that the length of the

review cycle becomes shorter. Instead of waiting for a time when several different people are available to schedule a meeting and review a new draft, the editor can send off a message as soon as the draft is done, and then each reviewer sends feedback as soon as they're available.

In a traditional workflow, if schedules are tight, the editor may have two separate in-person meetings, one with the director and another with the producer, who may give conflicting feedback. Asynchronous feedback allows each reviewer to respond to and discuss each other's comments.

Best practices for giving feedback

Notes. Comments. "Thoughts?" Whatever you call it, client feedback is part of the journey.

Perhaps you've seen some feedback (i.e. studio notes) on some famous films. Some of these notes, if applied, could've altered the course of film history forever. No doubt you've had your own moments where you've said, "Now… what are they trying to say here?"

Is the feedback experience meant to be confrontational or cryptic? How can you help your clients give meaningful feedback so everyone can do their best work? A specific review and approval tool might come to mind (coughframe.iocough). But before you bring in the machines, you have to address the human elements.

## Working with People

### Manage Expectations

You were hired as a problem solver. You're not solving The World's Problems, but you *are* helping to solve a ten-thousand piece puzzle with shot selection, scene rhythm, pacing, timing, and emotion, to name a few.

With that in mind, manage your client's expectations right from the start. What do they hope to achieve with this project? Who's the target audience? What should the audience take away? Information? A feeling? An experience? With the amount of time and money available, how far can you (or *will* you) take that vision? Confidently asking such questions opens the doors of respectful, candid communication throughout the project.

## Build Trust

Do you have access to scripts? Read them. Are there other preproduction assets available, such as storyboards, animatics, or previsualizations? Study them. Show them you can be trusted to ask, "What's next?" until the project wraps.

Have an idea how to take something in a certain direction? Toss it out to the client. Describe how it would take shape. If you share the client's vision, and can reach the desired outcome, you're likely to work together again. The more you work together, the more you'll be trusted to just take the ball and run with it.

## Build Rapport

Sometimes, it's a bit of conversation with the client. Other times, you'll have little-to-no pleasantries. How will you know which way to go? Read the room, then build rapport with the client.

Unfortunately, this is an art, not a science. Did the client show up with a smile and some pep in their step? Or are they dragging because of decision number nine thousand nine hundred ninety-nine? If the client's having a bad day, don't ignore it. Acknowledge it. Listen. Then, figure out a way to bring it back. Maybe it's a favorite jam. Maybe it's a classic YouTube clip. Maybe they just need some time alone. If you get it wrong, that's OK. Read the room. Organically build that rapport. Show the client you're not just a pixel pusher.

If you're not great at this skill, the best way to learn is by watching, rather than reading books. Try to find someone who's a pro and feeling the room and creating a connection with the client/producer/director, and see if you can be an assistant or intern for them.

## Own Your Expertise

More than likely, you were hired because… you're the expert. Own that expertise.

This doesn't mean being tyrannical. Or belligerent. Or acting with hubris. But if you made a decision, and they want to know why, your decision wasn't arbitrary. Confidently explain The Why. They may not agree right away. There might be some back and forth. That's OK. You're the expert. *Be* the expert. But…

## Be Nice

Be open. Plenty of good ideas start as, or come from, bad ideas.

Be yielding. Try something, show it to the client. If they don't like it, retreat. If you believe in it, pitch it again at another time. If the client still isn't feeling it, but you still believe in it, try it

one more time, at another time. After that, no more. If you want that client to *want* to work with you, yield.

Be respectful. If they're short on time, and they tell you, respect that and think about re-organizing how you run the session in order to take care of the most important questions first.

Be kind. As it's been said of feature film work, getting the job isn't about convincing someone you're the best editor around. It's about convincing someone that spending nine months locked in a room with you will be enjoyable.

## The Feedback Loop

### Using Review and Approval Services

The other side to getting meaningful client feedback is having the right toolset in place to do so. Working with the client in the same room at the same time makes this relatively straightforward. But when working across buildings in the same facility, or with team members from around the world, common feedback mechanisms frequently fall apart. Email is often used for project management. Something as simple as hitting Reply instead of Reply All, or having offhand comments misconstrued, could result in disaster.

Consider using a Web-based review and approval service that locks all feedback to the timeline. With it, everyone can have a unified toolset as a means of giving clear, focused feedback. Some can even transform that feedback into a usable component for your NLE. How?

With timecode as the anchor point, you can create a project where members to view a video, stop playback, and submit

comments attached to that timecode. Comments usually appear in a chat-style interface, which encourages realtime discussion. These comments could then be exported as markers to import into your NLE, which could checked off in your timeline, one-by-one.

Some services include drawing tools to help you zero in on a particular item. "I don't like the lady in the yellow hat." is tough to fulfill if there are several shots featuring ladies in yellow hats. Circling which lady appears at a specific timecode, or even drawing an arrow to her, gives that request some clarity.

But what if, despite best efforts, communication breaks down? Maybe you wrote something that was misunderstood, or you're not seeing eye-to-eye on a certain item?

## Discussing in Realtime

Hop on the phone. FaceTime. Skype. Slack. So much can be cleared up in realtime. With the tone of your voice and your facial expressions, incendiary moments can be diffused, confidence in decisions can be affirmed.

Keep it brief. Or not. Again, read the room.

Once you get the feedback needed, document it, then execute.

"Filmmaking is the ultimate team sport." You and your client are on the same team. Show them you can work with people, not just software. Help your client use the right communication tool at the right time. They'll deliver clear, meaningful feedback, and you'll lay the groundwork to be seen as, not another hired gun, but a trusted partner.

Contributed by Isaac Terronez, Producer, video editor, and technologist.

## Distributed/remote editing workflows

While it is now quite common for directors, producers, and clients to give feedback remotely, it's less common for the editorial team itself to be distributed across multiple locations.

It's commonplace across all genres of filmmaking for the different phases of post-production to happen in multiple locations. The edit may be in LA, the color correction in New York, and the VFX in Toronto. That is just as likely for a 30-second advertisement as for a 2-hour feature film.

Handing off a locked edit to the colorist is much simpler than working actively in tandem with another editor, however. The editors and assistants need access to *all* of the footage, whereas the colorist only needs access to the clips used in the final project. It's pretty easy to transfer 2 hours of footage to the colorist over the internet, but it's much harder to transfer 400 hours of footage to another editor.

On a smaller project, it may be quite feasible to send the media files to another editor over the internet. On a larger project with terabytes of footage, it's usually easier to just mail a hard drive.

The second challenge of remote editing is that everyone has to stay in sync. The assistants need to be able to see what the editor has been working on, and vice versa.

Fortunately, the project files, which hold all of the work done by the editor and assistants, are usually very small, and so it's easy to send

them back and forth or to use a web service to keep them constantly updated. All of the work done by the editorial team is contained in these small project files, so once everyone has access to the media files, there's no need to transfer them again.

The core idea of how multiple editors work together remotely is the same as how they work together in the same location: one editor "locks" a sequence or a folder, and the other editors work on other sections until the first editor has finished and unlocks the sequence.

"Searching" editors Nick Johnson and Will Merrick.

The major difference is that, when an editorial team is working together in the same office, they work off of a single set of shared files. This makes it simple to designate which files are locked and which are unlocked.

With a remote workflow, however, each editor must maintain their own set of files and keep them in sync at all times. It's vital for all users to maintain an active internet connection to keep their files in sync whenever they are working so that they know what is locked and what is unlocked. Otherwise, two different editors may begin working on the same sequence at the same time. Both editors will "lock" the sequence, but since they are not connected, that information is not communicated. Once sync is restored, there will

be a conflict, and one editor's work must be discarded.

On most feature films and TV shows, this type of distributed workflow is rare. It's still the case that the editor and the assistants work out of the same location (though often in separate rooms). New media companies and video marketing teams are expanding their video production capacities rapidly, however, and it's much more common for those teams to be distributed across multiple locations.

## Editing Hardware

As we mentioned above, if you are using an offline workflow, then the editor's computer usually doesn't have to be extremely powerful. The colorist and the VFX artist often require very carefully-tuned powerful custom machines in order to handle the enormous task of processing hundreds of millions of pixels every second for high resolution images, but the editor will probably be fine with a mid-to-high-tier standard computer.

It's quite possible to edit a film on a laptop, but a professional edit suite will allow the editor to work much more efficiently and comfortably, and so an edit suite usually requires several high-quality screens, professional speakers, a comfortable chair, and some high-performance storage.

Editing hardware

The NLE is the most important piece of software for the editorial team, and so it largely dictates which hardware you'll choose for peak performance. Editing effectively requires stability and performance between hardware and software.

Each NLE favors particular types of hardware, though there's always a range, and the more you spend, the better performance you'll get generally.

How can you choose the best hardware for your situation? It all starts with your NLE's system requirements. Here are some requirements for frequently used NLEs (presented in alphabetical order):

- Adobe Premiere Pro CC
- Apple Final Cut Pro X
- Avid Media Composer
- Blackmagic Design DaVinci Resolve 15

Looking through them all, you'll see some common threads.

## Core Hardware

### CPU (Central Processing Unit)

The first thing you'll notice? These all run on Intel-branded hardware.

The next thing? The clock speed. More gigahertz translates almost directly to faster performance, but you also need to decide how many cores. Multiple cores allow you to run multiple operations simultaneously, but not all operations can be spread across multiple cores. Some tasks do a great job of making use of 6, 8 or 12 cores. Other tasks will only use one core, letting the rest sit idle.

Add to that the size of the CPU cache (again, higher numbers

equals faster computing), and you can decide which combination is best for your budget.

Look for CPUs listed with these specifications:

- 6th Generation Intel Core and newer: i5, i7, or i9.
- Intel Xeon Workstation Family: E Series and W Series

## Storage

A high-performance editing computer should always have at least two drives:

The system/boot drive – internal, where your operating system lives.

The media drive – internal or external, where your editing assets live.

The software application and support files for your NLE are stored on the system drive. Project files, Libraries, and editing assets are stored on the media drive. It's important that these be separate because the software will often need to be reading from both locations simultaneously. If your project files and media files are on the same drive as your software and operating system, that will slow you down.

For peak performance, purchase a workstation with a SATA III solid state drive (SSD) for the system drive. When your operating system runs out of memory (RAM), it'll use start using portions of your system drive instead of memory. This is called swap. The faster the system drive, the less you'll notice when you've run out of RAM. (And you will).

Media drives are typically external drives (although a large

enough internal drive can be partitioned to achieve some of the same benefits of an external drive). For peak performance, you'll want to purchase a drive that has high sustained throughput and also has a fast connection (i.e. interface) on your workstation. If you buy a high-end SSD and connect it via USB 2, you're wasting your money.

## Memory (RAM)

Memory is often confused with storage, but they are very different things. Storage (hard drives and SSDs) is where you put your files. Memory, or Random Access Memory (RAM), is a temporary, high-speed staging area where a computer stores and retrieves calculated data. Once your computer turns off, the RAM is wiped clean.

So how much RAM do you need? In short: you can never have enough RAM.

Most Windows-based workstations have computer cases you can open up and upgrade later. Recent Apple offerings have become less and less user-upgradeable, which means that you need to predict how much RAM you're going to need when you buy a Mac.

If you're doing any kind of significant video processing, you'll want as much as you can afford.

## GPU (Graphics Processing Unit)

The GPU is a specialized tool that's optimized for dealing with images, unlike the CPU which is a general-purpose tool.

GPU advancements have outpaced CPUs in so many respects that NLE developers have taken advantage of that extra horsepower to offload tasks once reserved for the CPU.

GPUs come in three flavors:

1. Integrated
2. Discrete
3. External

*Integrated GPUs* are built into your workstation's motherboard. It's a budget-friendly option, and some NLE's can competently perform on these. But if you know you'll regularly work with full-resolution media or you'll be doing any online or finishing work, you'll need more power. Enter discrete and external GPUs.

*Discrete and external GPUs* are generally both more powerful and physically larger than integrated GPUs. A discrete GPU is inserted into an open slot in a desktop computer and thus cannot be added to a laptop. An external GPU connects via a cable and can be added to a laptop or desktop.

Just like with multi-core CPUs, it's up to the software in question to take advantage of the GPU's power. Playback or rendering functions that take advantage of these are commonly referred to as *GPU-accelerated*.

Just like CPUs, deciding on the right GPU will be a balance of expected performance to price point.

## Displays

For most editors, the most important feature of a display (besides its size) is the ability to display accurate colors. If you've ever walked into a Best Buy or a Walmart and seen 10 different TVs playing the same demo loop, you'll know that the same image can look vastly different on different displays.

Low-quality monitors can have strong color casts or may be unable to properly display colors at the ends of the visible spectrum. These can cause serious issues to an editor and are completely unusable for color correction.

These are some of the core factors to consider when searching for a high-quality display.

- IPS (in-plane switching) and OLED are generally known for good viewing angles and accurate color and contrast reproduction
- Good contrast ratio (ration between darkest darks and brightest brights)
- High resolution as more projects are shot in 4K, 6K, 8K and higher.
- Wide color gamut support (as wide as possible)

When correct color is absolutely essential to the final delivery (e.g. delivering media to a large distribution network like Netflix or Hulu, or to a broadcast channel with specific color specs), a professional monitor as well as a color calibrator will be used.

## Additional Hardware

### I/O Cards and Breakout Boxes

I/O cards and breakout boxes extend your computer's capabilities for specific purposes. Those include:

- Capturing and importing footage from an analog source, such as miniDV.

- Receiving and recording audio from an external source, such as a microphone with an XLR connection.
- Recording video and audio to an analog deliverable, such as Betacam SP.
- Powering extra monitors.
- Decoding raw footage more quickly (RED Rocket card)

Some NLE developers, like Avid and Blackmagic Design, sell cards or breakout boxes specifically designed for their software. Third-parties like AJA sell devices for use with multiple NLEs.

## Speakers or Studio Monitors

Sound is an extremely important element of any video. Even if the project will be handed off to a professional sound mixer later on, it's important for the editor to be able to monitor the sound accurately while editing.

Assuming you're not working with audio beyond stereo sound, here's what to look for in a proper set of speakers:

- Size – how much physical space is available?
- Active vs. passive – is the sound amplifier part of the speaker enclosure, or will you need a separate component to power the speakers?
- Frequency range/response – what sonic range can these speakers reproduce?
- Intended listening distance – are they designed for near, mid, or far listening distances? More than likely, you'll choose speakers meant for near-field listening.

For best listening, form a triangle with your speakers, point

both of them at your main sitting or standing position. You'll be surprised how much you miss when they're just facing forward.

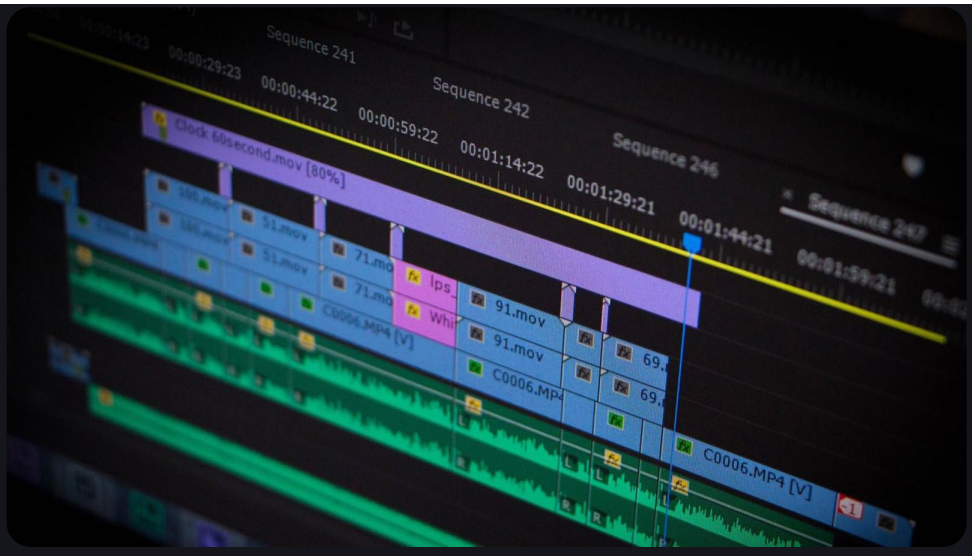
Choosing hardware that's the best fit for your day-to-day NLE will keep you focused on your story and not on IT. Those choices may also affect how much support is rendered by the vendor and fellow users when things go awry. So choose wisely, and you'll keep that edit going at the speed of thought.

Contributed by Isaac Terronez, Producer, video editor, and technologist.

Share storage is of course one of the most important pieces of hardware for the editorial team, as we mentioned above.

## Editing Software

The NLE (non-linear editor) is the primary tool of the editor, and each of the main NLEs provides all of the functions needed to complete an edit, whether you are cutting a full-length film or a music video.




The core idea of what an NLE provides is very simple: organizing clips, creating timelines, and then cutting and moving clips around in the timeline. Each NLE adds dozens of specialized tools which make those processes quicker, easier, and more manageable. Those tasks sound simple, but when you have hundreds of hours of footage and thousands of clips on your timeline, you really need those specialized tools.

Each of the core NLEs also offers a suite of tools to accomplish the other main branches of post-production: color correction, sound editing and mixing, VFX, and delivery. On high-end projects, each of those tasks is handed off to a different expert who uses dedicated software. With lower-end projects or fast-turnaround projects, however, the tools that are built into the NLE may get the job done.

NLEs

When it comes to professional NLEs, there are four clear leaders: Avid's Media Composer, BlackMagic Design's DaVinci Resolve, Apple's Final Cut Pro X, and Adobe's Premiere Pro.

All four of these are excellent tools fully capable of handling almost any editing task. That said, each tool is different, has its

own history, and is best suited to particular types of projects.

We've identified a few major areas of differentiation.

## Traditional vs Nontraditional Layout

Avid, Premiere, and Resolve all work on a common and traditional paradigm. Video tracks, audio tracks, bin organization, and similar naming conventions. Apple's Final Cut Pro used to also fit nicely into this description, up through version 7.

In 2011, Apple rewrote and reimagined the NLE from scratch, building the tool around a very different paradigm from the traditional track-based and bin-based tools. It was new, it was different, and when it originally came out, it was lacking many professional features.

The release and complete rewrite of Final Cut Pro X upset a lot of editors (this one included) and created a very emotional divide among some professionals. Some will be very passionate about their views. In the years since its release, the missing features have all been filled in (and then some), bringing FCP X back up to the professional level.

I have become a FCP X user and used it at the highest level, a major studio feature film. It is a valid, viable and proven tool.

It is, however, different from the other three in its core design. While Avid, Premiere, and Resolve use tracks as the primary way of arranging clips on a timeline, FCP X uses a magnetic timeline to avoid tracks entirely. While this may seem like a small difference, it marks a deep change in the way that the editor works.

## Cost

Let's look at this from the perspective of an individual, rather than a company with negotiating power and lots of resources. Depending on your use case, you have several options to consider: full purchase, subscription, or free.

All of these NLEs are capable but have different strengths. For example, are you working with many editors (picture editors and sound editors) attached to shared storage? Media Composer is the first choice for that in Hollywood. For a project with very complex organizational needs or for projects cutting on camera original footage (not proxies), Final Cut Pro X is a great choice. For deep integration between apps, consider Adobe's Premiere Pro as part of the Creative Suite. For industry standard color controls and a tight post-production pipeline, try Resolve, which has editing, color, sound mixing and nodal visual effects built in.

All that said, here's the basic price of each at the moment.

Avid Media Composer: from $200 per year to $1300 for a perpetual license.

Final Cut Pro X: $299. Upgrades have been free since the original release of FCP X.

Adobe Premiere Pro: $20-$80 per month, depending on your subscription.

Davinci Resolve: Free for single users, $299 for a more capable version that allows multiple users.

## Ease of use

These tools have somewhat different learning curves. Because people generally prefer what they are used to, solicitations from your friends or other experienced editors may take you down a biased path. Thankfully there are free demos or versions of all of these NLEs. As well as many free and paid tutorials to allow you to try before you buy.

My opinion is that editors experienced with a track-based editing system will find that type easier to use and easier to switch between the three track-based NLEs mentioned here. For a newcomer, I think Final Cut Pro X is easier and more intuitive.

Many professional editors, who are highly accomplished and skilled and technically savvy (editors who have used track-based NLEs) find FCP X flummoxing. When I approached it as a professional, I did, too. I actually asked for a refund! But eventually, I gave it another shot. I asked a lot of questions because it seemed to be too easy, too simple and less powerful. After I learned it and understood its logic, I found it, and still find it, very easy to use, deeply powerful and doing a great job of staying out of the way of the creative process.

## Multi-user workflows

While it's possible to use any of these tools with a team of editors, some of them make it easier than others.

Avid is certainly the most widely-respected in that area, followed by Resolve. Adobe's recent releases have included some multi-user workflow options, though they aren't yet as widely-used as Avid's.FCP X as of yet has no built-in solutions for multi-user workflows. It is certainly possible to {work

together with multiple editors in FCP X}, but it's not as smooth as with Avid.

## Quick Overview of Each Tool

Avid Media Composer – The king of collaboration, standard in Hollywood, stable tools. Learn it if your destination is Hollywood feature films and television.

Final Cut Pro X – Fast and flexible, easy to use, very friendly toward original media, one person post-production house, great integration with many utilities thanks to advanced XML. Learn it as a cost-effective, high-performance, one-stop shop for editing, color, visual effects, and sound work.

Premiere Pro – Well integrated with Adobe's Creative Suite, powerful color and sound operations, tight visual effects integration with After Effects, widely used. Learn it as an effective tool that has been embraced by many in television and film.

Resolve – A promising newcomer, even at version 15 I'm still calling it a newcomer! Top-tier color tools, fully integrated workspace with editing, sound, visual effects, color and detailed delivery options built in. Blackmagic Design is being very ambitious with this software and I'm looking forward to seeing both how it works and how it affects the industry. Also, the single user version is free! Learn it if you want to be ahead of a possible coming wave.

## Your own use case: the real question.

Are you working alone? In a group of editors? With a design

team? Are there a lot of visual effects? How do you plan to finish your project?

There is a myriad of scenarios.

For example, if your favorite visual effects artist uses Fusion, DaVinci Resolve is a good choice. If they use After Effects, maybe consider Premiere. If you have Sound Designer who works on Pro Tools who requires detailed film style change notes, Avid is a good choice. If you're shooting on a RED camera and want a simple flexible way to cut with your original camera media, Final Cut Pro X is a great choice. Are you working alone? Plan to do editing, color, sound, and visual effects yourself? Premiere, Final Cut Pro X and Resolve will do the job!

The more modern editing systems (Final Cut Pro X, Adobe Premiere and Resolve) support, in more convenient ways, modern workflows and elements like original camera media, original sound media and mountains of metadata which have great value for visual effects artists, sound editors, and colorists.

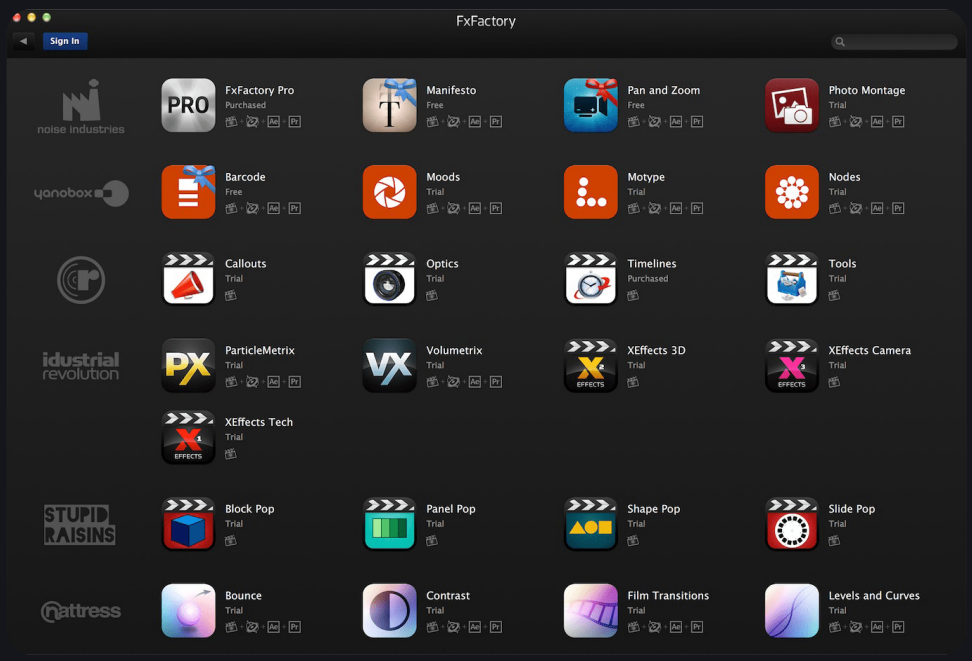
There is no best NLE for everyone.

Contributed by Mike Martzdorff, Editor, Warner Bros.

There are also dozens of tools and utilities that exist alongside the NLE, which can simplify particular tasks or add more advanced features in a particular area.

Every NLE provides color-correction features, for instance, but not at the same level as purpose-built color tools. So it's common to add extra color-correction plugins which may offer more advanced or more intuitive options. Or, if you're using Frame.io for remote feedback, you may add the Frame.io plugin to bring review and

collaboration tools into the NLE.




Plugins like FxFactory by Noise Industries provides an entire ecosystem of third-party plugins for post-production software from Apple, Adobe, and Blackmagic's Davinci Resolve.

## Common NLE Plugins

Lately it seems like every application platform is attempting to do everything all at once. Between Resolve and Final Cut X and their focus on doing everything from within one application, and Avid and Adobe trying to create full-fledged ecosystems within one brand, there is a big focus these days on decreasing, rather than increasing, the number of applications that a filmmaker needs to keep current.

However, no matter how hard software developers try, it's impossible for them to satisfy every need. There are just too many things that can be done, and too many different use cases, for a single product to cover it all. If nothing else, it's a matter of economics. Apple says there are more than 1 million installations of Final Cut Pro X. Development on that product

will inevitably focus on features that benefit the vast majority of those users.

But if you have an idea for an application that might benefit only 10,000 users, there might still be a real benefit in creating it, even if it's not worth it for the major companies to solve the specific problem you are struggling with.

That's where companion applications, helpers, and plugins come in. These are little tools that help make the post works job easier without necessarily being useful enough for everybody to immediately embrace. In fact, this space is not just one that is useful to filmmakers, it's also useful to the developers of the major heavy hitter applications as a proving ground for new technology.

## Breaking new ground

For a long time we had to manually sync our dailies by hand, unless we were working on a timecode sync based production. It was a laborious process. Then along came Pluraleyes, initially a standalone application and a plugin for your editing platform that used the waveform of the audio in your video file to sync up picture to the waveform of your audio recording. All of the sudden something that used to take hours of hand labor became an automated process. Not "instant" like timecode sync, but less laborious.

It was a revolution, and it's still the best example of the type of innovation that often comes from smaller developers thinking about problems in new ways. So what happened? Within a few years, every major editing platform had integrated a native waveform syncing feature, and PluralEyes was purchased by the plugin maker Red Giant, who have continued to innovate with the product to provide functionality that might be missing from the native tools.

Not every innovation that comes from a small startup gets absorbed by the big companies. Many functions just aren't so useful that they are a natural fit for the bigger platforms. It's important to remember that just because a certain major platform has a given functionality doesn't mean that it has put all of the work and innovation into making that function as powerful, or easy to use, as it could be. For instance, Da Vinci Resolve has an excellent noise corrector built into the platform, but it is not uncommon to see other powerful noise correctors installed in the software in various post suites since they might offer real power on particular shots, or simply because a given editor or colorist is used to it.

Plugins remain useful even when there is a similar functionality in the full NLE, generally providing power and interface optimization that are the result of a design team focused on one thing and one thing only. The folks at Neat Video do noise correction, that is it. That's their jam. There will always be benefits of that.

## GPU Optimization

One other area where smaller vendors often shine, especially in the last few years, is in accelerating their workflows. For the longest time, it was the default assumption that any time you added a plugin the timeline color would change to "needs render," since plugins tended to be very processor intensive.

However, the major manufacturers have worked extensively to create GPU optimized plugins that are able to create quite stunning effects, often in real time, meaning they can immediately play back at full speed without rendering. This speed opens up new avenues for creativity, since it's easier to explore when you aren't constantly waiting for everything experiment to render to see how it looks. In fact, GPU

acceleration is an important enough aspect for plugins that Avid Media Composer has created a special symbol, the green dot, to allow end users to easily see what plugins are GPU optimized.

Before we talk about film industry specific plugins, however, there are a few applications that weren't built specifically for filmmakers, but it is worth it for most filmmakers to know about them. The vast majority of modern filmmaking software (with the exception of Final Cut Pro X, of course) is platform-agnostic, meaning you can run it on Mac or PC easily, but many of these outside applications tend to be Mac-specific, at least to start.

## The Big Players In Plugins

Red Giant and Boris Effects

There is no real way to talk about plugins and helping applications without talking about the two biggest vendors in the field, Red Giant, known for Magic Bullet, Denoiser, Colorista, and Boris FX, who own Sapphire and Mocha along with Continuum.

Why are these two companies so dominant? Well, aside from both making great products (they do!), there is a natural tendency of this market to consolidate since it makes for easier licensing at the corporate level. Let's say you are a medium-sized cable TV network with 100 editors in your facility; do you want to have separate contracts with 20 different plugin makers for all 100 seats in your facility? Or do you want to have a single vendor offering you all the functionality you need within a single license agreement?

While the space for the "individual editor buying a single

plugin" remains highly competitive with players big and small, the market at the corporate level is very dominated by the major vendors because of the customer service support they provide and the integration with a larger facility.

This means it's smart for you to get to know the offerings from these players since it's likely you'll sit down to a system with these installed on your next freelance job. Take a look at the plugins installed on any system where freelance on since you never know when it'll be useful and it's nice to know it's already there instead of frantically hunting at 3 am looking to see what tools might be installed.

Facility licensing is especially important since it allows for users to move projects around from machine to machine effortlessly. If you've ever worked on a project in a school or facility with plugins and layered many of them on, then gone home only to discover a massive red "Plugin Not Installed" warning when you launch, you know how frustrating it can be to get back on that same machine, or have to purchase the plugins you need for a home system, just to finish the project.

One tip is that, if you are working somewhere with plugins and know that you will be leaving that facility and finishing elsewhere, you should consider "baking in" your plugins. This means rendering out each shot with a plugin on it to a fresh video clip, then re-importing it "baked." This can be laborious, and of course you lose the power of raw video, but it can also be well, well worth it.

Conveniently, both Red Giant and Boris make it very easy for students to get their hands on their tools, since of course if students get used to/addicted to the tool in school, they are more likely to push their bosses to pay for the full license later. Boris has a program that gives a license to anyone with a .edu email address and a student ID, and Red Giant allows for combined discounts, which is exceptionally rare. So, for instance, you can apply BOTH your student discount AND any

sale they have into one massive super discount. Every December they have a massive sale, and if you are set up with a student discount early you can potentially get 80% off permanent licenses of the software.

Just because these two are the biggest, however, doesn't mean they are always the best, and when you are considering plugins, it's well worth testing out others.

## Looks

By far the most common arena for plugins is creating "looks", taking "bland" footage and adding spice, flavor, or personality. With look names like "Hollywood" or "Cinematic" or "CSI," the purpose of these plugins is generally to take video footage and make it feel more expansive.

Of course, most projects go through a dedicated color grading step, so the color grading plugins are less prominent than they used to be. However, when the deadline is tight, when you might not have the full color grading skills, or you just want to experiment, they are worth considering.

Red Giant Magic Bullet Looks and Colorista and Boris Continuum are the popular choices, with Hitfilm from FXHome being another popular choice.

## Titling

Another major arena for plugins is to bulk up the titling sophistication of an editing application. While bigger projects continue to use dedicated motion graphics tools, most likely After Effects, sometimes you simply don't have the time or

manpower to manage the round trip, and you want more sophistication or "pizzazz" than is offered in the native editing titling tool.

Titling is another area where plugin makers have either been absorbed by, or created long-term licenses with, the major NLE makers. Media Composer ships with the New Blue Titler, licensed by Avid to enabled\ more sophisticated titling tools. Boris Title Studio and the Red Giant Effects Suite are also popular options.

## Tracking

Sometimes you need to track a specific element in frame. Maybe you want to replace a TV screen in a handheld shot, or you want to place some text so it feels like it's part of a shot. Tracking is an essential part of that process. This is an area where the current leader is clearly one product, Mocha.

Mocha created ripples through post when it first became available for its amazing planar tracker. Mocha is so good, in fact, that a toned-down version of Mocha is licensed by Adobe to come standard in After Effects. However, there are standalone and integrated versions of Mocha available for all the major platforms, and the toolset continues to constantly evolve, with a growing set of VR tools coming out in recent years.

## Special Purpose plugins

Beauty Box

Beauty Box, from Digital Anarchy, is a specific toolset

designed purely around "beauty work," the process of de-aging or cleaning up the skin of a subject. While many associate this work with cosmetics and fashion, it is just as common in narrative and documentaries these days, and the beauty box toolset highly targeted at making it fast and easy for the user.

Noise Correction

Low light video noise is one of the major frustrations of filmmakers everywhere. Maybe the camera was slightly too deep in stop, or there just wasn't enough light on the scene, but you didn't have a well-exposed image and there is a fine, dancing grain on it you want to remove.

While there is a native noise correction built into Resolve Studio that is quite robust (though very GPU intensive), the general solution most pursue is a plugin, with Red Giant Magic Magic Bullet Denoiser being a very popular solution, especially when bought bundled with the rest of the magic bullet suite.

But Neat Video is worth consideration as well. It plugs into all the major platforms, and Neat Video will sometimes help a shot when you felt completely helpless.

Knoll Light Factory

While the film industry is less lens flare addicted than a decade ago, lens flares are still a powerful way to elevate and transform a shot, and the powerful customization tools found in Knoll Light Factory make for a fun, flexible, lens flare-tastic experience.

## Interface Plugins

One final arena of plugins worth remembering is plugins that are designed to help a user interface with an outside platform more seamlessly. Online customizable stock footage platform Filmstro has a plugin for integrating their tracks into your timeline seamlessly. Even Shutterstock has a plugin for shopping for and licensing your stock footage.

Frame.io has a great plugin for integrating client notes to a timeline. Rather than having to go back and forth with a browser, checking every note and bringing it up to the timeline, client notes are brought in to a timeline as markers, making them easier to navigate to and check off the to-do list.

## Helpful Tools

Amphetamine

Amphetamine is a menu bar application that stops your Mac from going to sleep. That's it, that's all it does. But rather than having to go into your system preferences every time you are doing a long file transfer or render and you want to be sure it makes it, you can just head up to the menu bar and click on the little customizable "Amphetamine" icon (we prefer the coffee carafe to the pill, it feels more businesslike), and your Mac won't sleep.

It does just what you want it to do. There is a similar app from Zhorn software for PC called Caffeine, though we haven't tested it to know if it's as reliable as Amphetamine for Mac.

Activity Monitor and iStat Menus

Most Macintosh machines come with a great little program included called "Activity Monitor," which can be super helpful as you try and calculate various situations and see whether or

not something in your setup is causing you to lose valuable time on a process.

Worried if having a few tabs open in chrome is slowing down your render in Premiere? You can watch activity monitor for CPU activity, and it should give you a sense of whether you are clogging up your processing power. Of course, you never want to use 100% of the power available; you need some overhead, but if the premiere render is only using 20% of your power, you have room to browse Buzzfeed.

Unfortunately, Activity Monitor is somewhat limited, and most users end up spending the money to purchase iStat Menus. Like Activity Monitor on steroids, it allows you to monitor all of the temperature sensors on the machine, see what processes are consuming what resource, and best of all monitor your GPUs.

Have a render constantly crashing? Check the temperature to see if you need to cool your system down, perhaps by putting it right by the AC vent or clearing space around it. Have a GPU installed but not seeing it in iStat? It might not be used at all.

SMC Fan Control

SMC Fan control lets you control your fans. You Mac is generally trying to keep your Mac cool, but also quiet, and will slow your fans when possible to avoid making noise. As a filmmaker with a big render and a deadline you generally want the fans cranked to the max, and can live with a little buzzing noise. With SMC you can crank your fan speed to the max, to pre-preemptively get cooling your system before a big render.

Blackmagic Disk Speed Test

The computer industry tends to focus a lot on CPU speed in their marketing, largely because it's a simple number to try and benchmark. But for many tasks filmmakers regularly embark on (camera downloads, renders), disk speed can be one of the huge bottlenecks. While Thunderbolt offers a maximum of 40Ggb/s transfer speed, that doesn't matter at all if the hard drive you are writing to can only read 125Mmb/s.

Blackmagic Speed Test emulates a common filmmaker scenario and tests drive speed (it will even tell you want resolution and codecs a drive will support). Next time you are wondering what the bottleneck is, think about drive speed.

Paragon

PC and Mac hard drives still don't play with each other natively. Worse, neither platform has native support built in for the Linux file format EXT, which is the required format for DPC drives (though unofficially NTFS is somewhat common for DCP delivery). To make one work with each other, the standard is Paragon. Affordable, stable, useful, chances are you'll be purchasing a Paragon application at some point in the future when the client brings in a drive in the wrong format for your facility.

DCP Transfer

DCP transfer is one of those applications that do one thing (transferring DCPs from one drive to another), but it does it well. Built on top of licensed Paragon drivers, DCP transfer copies and validates the mapping for your DCP, giving you peace of mind as you transfer your DCP from one drive to another. DCP is a notoriously finicky protocol. It is indeed possible to break it when copying, and DCP transfer provides real peace of mind.

Contributed by Charles Haine, Filmmaker, writer, host and professor.

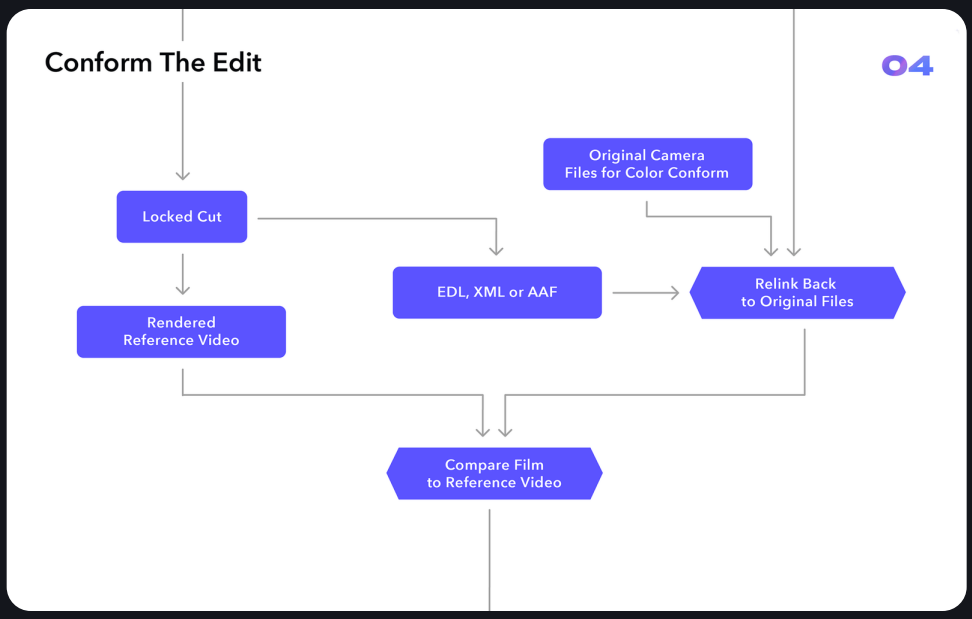
The assistant editors, who have to do a lot of checking, verifying, and organizing, will often use some other tools to help themselves stay organized and to allow them to send data between different applications.

CHAPTER 04

# Conform

The conform translates the film from the language of the editor to that of the color and sound teams. The tiniest details make or break this stage.

*7 minute read*




As soon as the edit has been locked, it's time for the conform. The conform is the process of transforming the project from a format that the editors use to the format that the sound, color, and finishing teams can use in their software.

The conform is when any organizational mistakes will show up, especially metadata errors, which can cause serious headaches, so make sure to keep track of that metadata!

A conform is necessary for a mid-tier or high-end project that will be sent out to another company or team for color and sound work. As we said above, it is the process of preparing the film to be sent to other teams/software. So, if you are doing all your work in a single tool or connected suite of tools, you can skip this entire section (apart from the relinking stage).

If you *do* need to send your project from your NLE to another piece of software, however, you have a series of steps to follow.

1. Export a reference video
2. Relink back to the original camera files

3. Simplify the timeline drastically
4. Export the timeline into a new format
5. Import into the finishing tool and check for errors

It's unusual for one person to perform all of these steps, and the order may vary slightly depending on who is doing the conform. In some cases, the editorial team may do most of the conform, but it's more common for most of the work to be done by the colorist or finishing artist who is receiving the project.

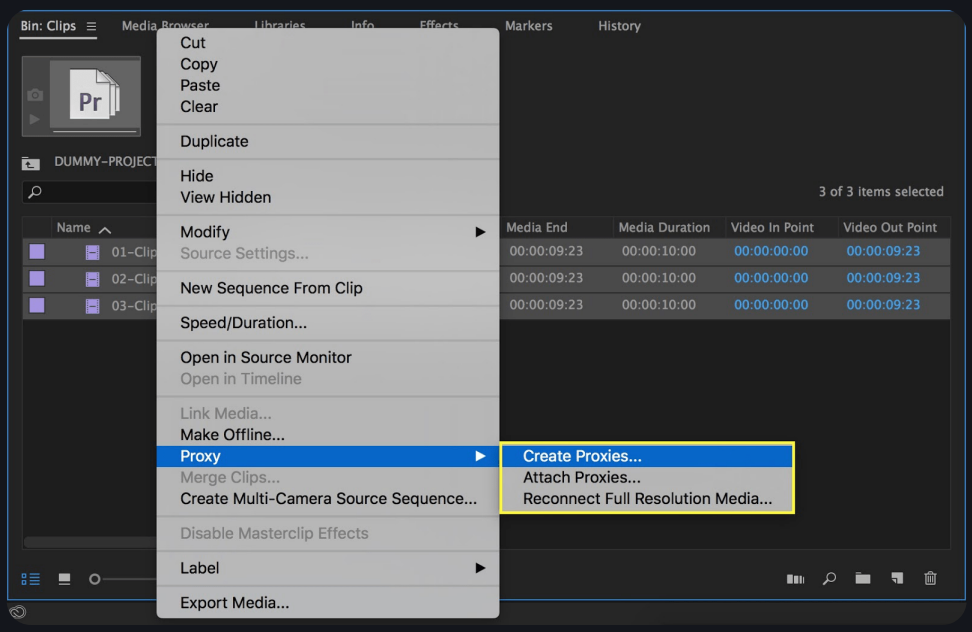
## Export a Reference Video

The first step in the conform is to export a reference video of the entire project as it was in its locked state. That is important because we're about to do a whole bunch of moving, swapping, and exchanging files. The reference video provides a check later on to make sure that everything was transferred properly.

## Relink back to the original camera files

Most people these days use an offline editing workflow (and the ones that don't, probably should), which means that they were editing off of lower-resolution proxy files.

Those proxy files are great for editing, but they're very low quality compared to the original camera files, and the colorist and VFX teams need access to all of the data that was originally captured. Even if you're doing all of your color and VFX work inside your NLE, you're still going to relink back to those camera files before you export.




Depending on what tool you're using and how all of your files were named and stored, the process of relinking could be as simple as clicking one button. Or, it could be many long hours of manually comparing files, one at a time. If you're consistent with your file naming, however, it can almost always be automated.

Proxy workflows in FCP X

Proxy workflows in Premiere Pro

Proxy workflows in Avid

These three guides contributed by Sofi Marshall, Freelance Editor.

Proxy workflows in Resolve

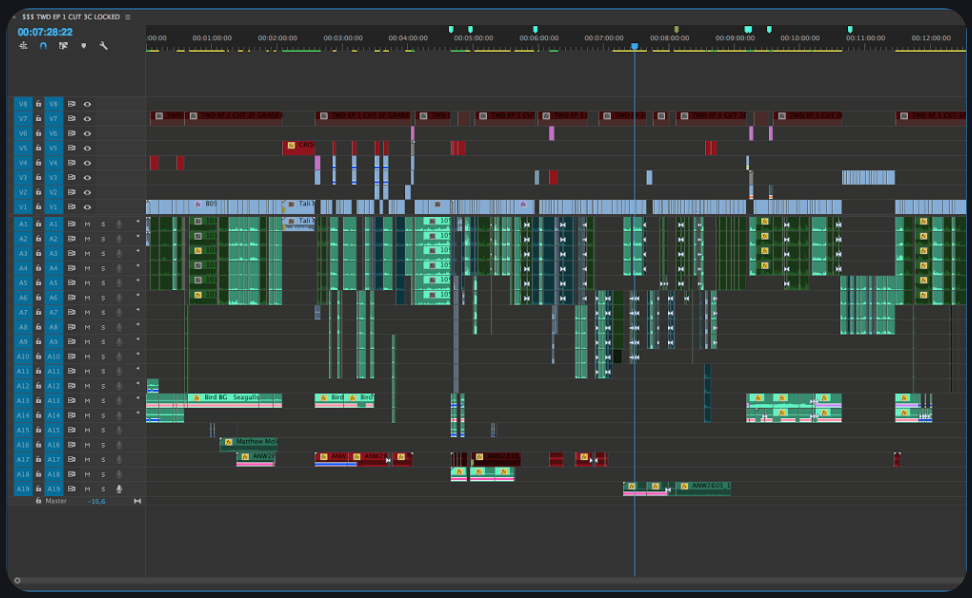
Contributed by Seth Goldin, Tech, post-production, and workflow at Freethink Media.

If your sound was recorded separately from your video, you also have to relink your audio back to the original files from your audio recordist. When you created proxy files for the editors to use, the camera files and the audio files were merged. Each clip in the finished timeline thus has to be matched to two different original source files—one from the camera, and one from the sound recorder.

## Simplify the timeline

Most interchange formats unfortunately can't handle much complexity in the timeline, and so we need to simplify the timeline down to the most basic setup—a series of clips, one after another—before we export.

Nearly all editors edit with a multi-track timeline, or else FCP X's magnetic timeline. In either case, it's common to have a dozen or more different layers of video. Imagine that the different video clips were stacked on top of each other. Whichever one is on top is the one that will be seen.




In order to prep for the conform, the film must be collapsed down

to a single track.

We also have to remove most (if not all) of the effects and transitions on our clips, since those effects and transitions will probably not translate to the software used for color and finishing. The finishing artist will have to recreate each one in the new software. This is another reason why it was important to export out a reference video, because the finishing artist will refer closely to that video file in order to recreate all of the effects and transitions that the editors used.

## Export the timeline into a new format

You would think that, by this point, we would have a universal format that everyone could use to exchange edit timelines between different software tools, but alas, one does not exist. There are several different competing ways to export a timeline from the NLE, but none of them is perfect.

Why there is no perfect timeline format

It might seem that a timeline is a very simple thing to store in data—it's just a series of clips, one after another, right? Why then are there data compatibility issues between programs?

One of the main reasons why is because a timeline actually contains a lot more data than just the order and length of clips. There are dozens of properties and effects that can be added to clips in a timeline, and each piece of software has its own options.

You may have added a blur effect, for instance, to a clip. There are dozens of different ways to calculate a blur, so if you send your project from FCP X to Resolve and tell Resolve to add a blur to that clip, Resolve's blur may look a little different from FCP X's blur, depending on which blur settings you used.

Or you might have added a plugin to your copy of Premiere Pro, and that plugin may not be available for Media Composer. In that case, it's impossible to send that project from Premiere to Media Composer and keep it looking exactly the same.

So while timelines seem simple, the sad reality is they are not.

EDL

An EDL, or Edit Decision List, is the oldest and simplest format used for exchanging timelines. The advantage of the EDL is that pretty much every piece of software can read and write EDLs. The downside is that the EDL format is extremely simplified. It cannot hold two video tracks, for instance, and most editors rely on at least half a dozen if not more tracks when editing.

XML

XML is a much more flexible format, and so it's possible to include much more data in an XML file than in an EDL file, but this actually creates another potential issue. Because XML is so flexible, it's possible for different tools to create XML files in different ways. An XML exported from one piece of software is not guaranteed to work in another. Some companies have created their own specifications in order to standardize how the XML is written (Apple's fcpxml, for instance), but they haven't been adopted universally. Some independent 3rd-party developers have created tools that help to translate projects between different tools, and some of them work quite well, but it's always a bit of a messy business.

AAF

The AAF format is similar to the XML format, but there is a governing body that helps to standardize how information is stored in the file. In spite of that, however, not all AAFs are cross-compatible with different systems. AAFs can contain both video and audio information, but they are especially common when sending projects to the sound department.

Both with XMLs and AAFs, it's necessary to research and test the workflow before you begin post-production to ensure that the project can be accurately transferred between the tools you are using.

Full Video Export

Another common way to send a film to the color and finishing departments is to export the entire film as a single video file. That file is sent to the colorist, who then has to cut it back up into the individual clips in order to color them. Color correction tools usually have the ability to auto-detect where cuts are and split the film up automatically. Those tools aren't perfect, however, and so it's still necessary to check the film closely and compare it to the reference video.

If you are using this method to send the film to another team, it's essential that you relink to the camera files before this step, otherwise you'll be sending a low-quality file.

Also, it's very important that you understand the codec, color space, and bit depth of the source material to ensure that no color information is lost during this export. All of the specifications of the exported video must be equal to or better than the source footage.

Choosing a codec to send to color correction

Coming soon!

Premiere Pro has a useful tool called Smart Render that can come in handy at this stage, if you are using one of the compatible codecs. Instead of rendering and exporting the clips on the timeline, it copies the data straight from the source files into the export file, keeping the color information untouched.

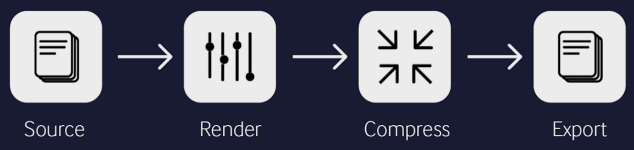
Smart Rendering

## Exporting = Rendering and Encoding

When Premiere is going to export some edited video, it has to do two separate tasks: rendering and encoding. It's important that you understand what that means, so you can understand why the below method can save you time.

When Premiere exports a video, it starts by reading the source file that you have imported into your project, which is nearly always a compressed file. Then it decompresses the file so

that it can deal with the actual image. It's not saving the uncompressed image to a file – it's just holding it temporarily in memory.




Rendering and Encoding for Export

Next it applies any transformations that you've done in your sequence – color correction, resizing, any manipulations to the image. That's rendering.

Then it encodes the image into whatever codec you chose in your export settings, and it saves that compressed data to the output file. So, we start with a compressed file, and we end up with a compressed file.

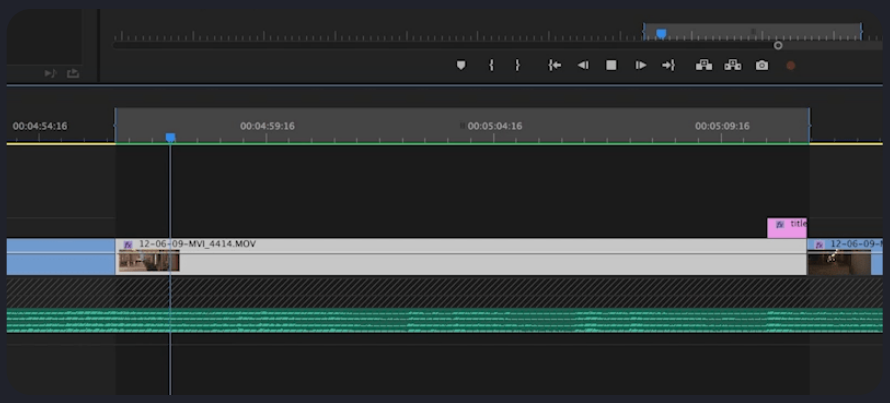
Now, it's usually not possible to skip either one of those steps –you usually have to render and then encode when you're exporting. But our first trick allows you to avoid rendering and encoding multiple times. In most situations, you don't just render out your timeline once. You often render out a rough cut, and then you get feedback from your producer or your client (perhaps using a tool like Frame.io), and then you need to export again after you have made some tweaks. And then the process repeats itself, so you often end up exporting the same timeline several times. If we're careful about our rendering and exporting, though, we can save a lot of time.

## 1. Compress Previews

Imagine you have a clip on your timeline with a bunch of color corrections. If you tell Premiere to export, it's going to render that file with your color correction, and then it's going to compress that data into an output codec. If it's a long clip with a lot of adjustments, it might take about 60 seconds.

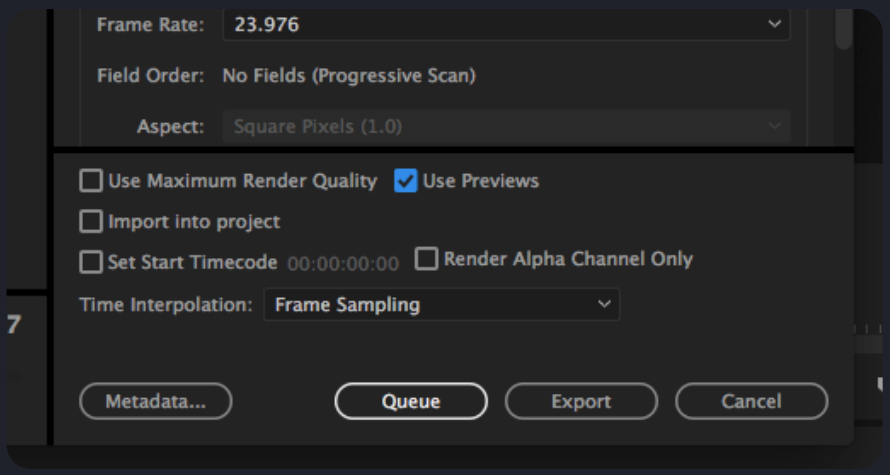
But you can also tell Premiere to just render the clip, and save that rendered file on the hard drive. That file is called a preview file, and you can use that file for exporting later without rendering all over again.

To start, click on "sequence" and then "render in to out". Now Premiere renders that clip and saves it away, and the bar on your timeline turns green.

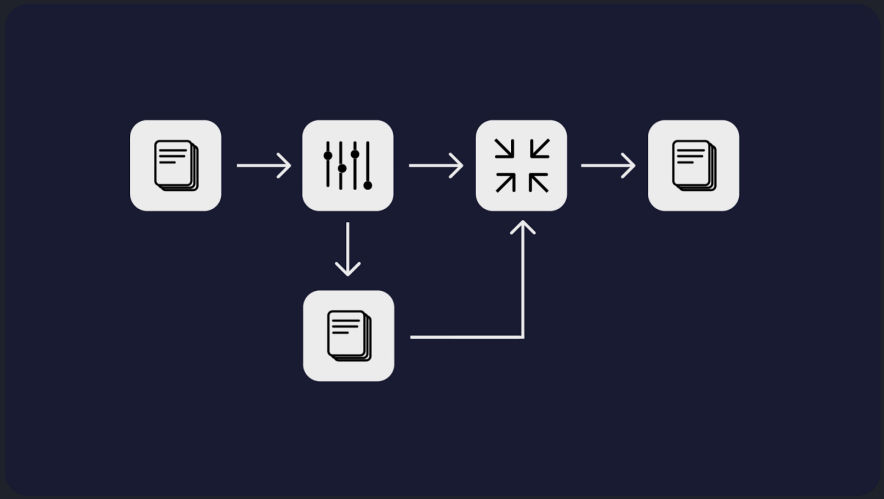



The green bar means that Premiere does not need to render that section of the timeline again when it exports.

Now if you go to export the clip again, and check the box in the export settings that says "use previews," now Premiere will simply take the file that it saved when you render and compress that to an output codec. It does not need to render any of the color corrections – it just needs to read that rendered file and encode it (we usually call that a transcode). And that takes only a couple seconds, instead of 60 seconds from before.




Now let's say that you want to change the title at the end of your film because there was a typo the first time. Premiere will still need to render the changed title, but it doesn't have to render the main clip again, which means that the second export will be almost as fast as the last one.

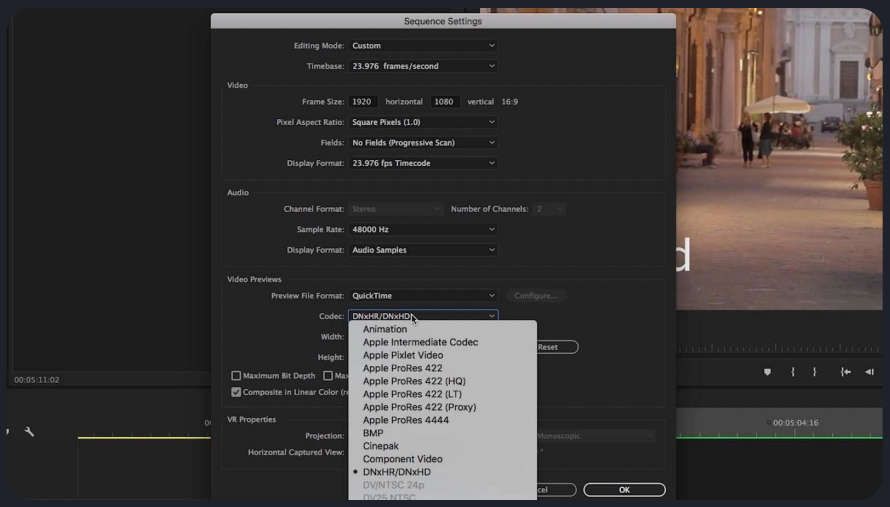


If you are only making fairly small changes to your edit, all of your exports from now on will be very quick.

All of that sounds wonderful, but there is a catch. By default, Premiere uses a fairly low-quality codec for its previews. That means that, if you check the box to use previews when you export, the final result will not have the full quality that you're expecting. That is fine if you are just going to export a low-resolution preview and it's not the final version, but what if you

want to export the full quality clip multiple times?

In that case, you need to change the preview codec to a high-quality codec, which is very easy. Go to “Sequence Settings,” then “Preview File Format.” If you want to be absolutely sure that you aren’t losing any quality, then you can choose a lossless or uncompressed format. Bear in mind that, if you use a very high-quality, high-bitrate codec for your previews, that may take up a lot of space on your hard drive, so most people will pick something on the high end but not the very top (maybe ProRes 422 HQ).

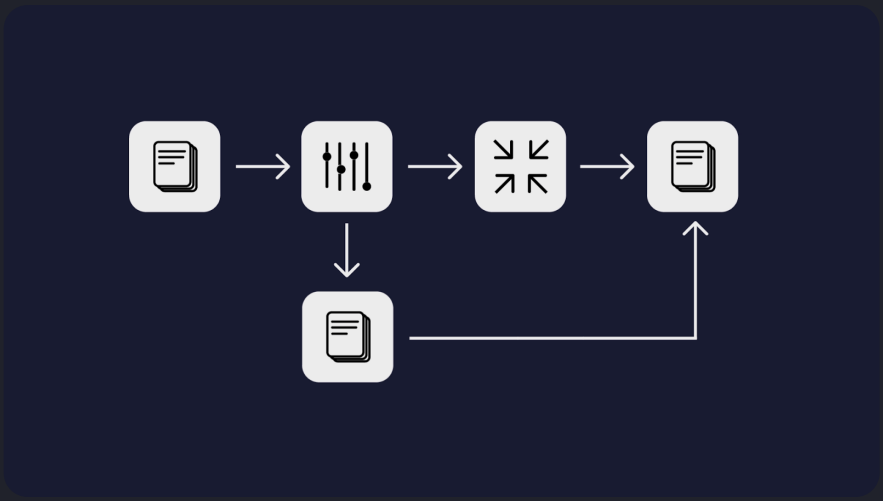



## 2. Smart Render From Previews

There’s one more feature of the “use previews” box that can make a huge difference in render times, and strangely, it’s seldom mentioned. If you choose exactly the same codec settings for preview files and for your output file, you can skip both the rendering and encoding when you export.

Because Premiere has already rendered the previews and encoded them, all it has to do is copy the data into the output file. Premiere calls this Smart Rendering, and in this case we are actually combining the Use Previews feature and the

Smart Rendering feature.



Check out the video for a detailed walkthrough.

That clip, which took about 60 seconds to export normally, and took about 5 seconds with the normal "Use Previews", takes less than a second when you Smart Render from previews.

This trick has the extra advantage that the image is not processed twice. If you use previews without the Smart Render, then your image will be compressed twice (unless you choose an uncompressed codec as your preview codec, of course).

This trick doesn't work for all codecs unfortunately–you have to use one of the codecs listed under "QuickTime Match Source (Rewrap) Support". The two most common codecs listed are ProRes and Cineform, both of which are great.

The tricky part is knowing whether or not Premiere is using your preview files exactly as they are and just copying them into the export file, or whether Premiere is reading those preview files, decoding them, and then re-encoding them into the output file. It's important for you to know, because if it's re-encoding those files, then you could be losing some

quality, because the images are being compressed twice. And of course it will take longer, which is lame.

Fortunately, it's not actually that hard to do. Select a clip, pre-render it, and then export with the Use Previews box checked, and write down how long it takes. Then export again, but make a small tweak to your output codec. You should use the same codec, but make a small change (maybe go from ProRes 422 to ProRes 422 HQ). You should see the slightly tweaked export taking longer, because now it's having to transcode your preview files, whereas the first time it just copied the data. If it takes about the same amount of time to do both exports, that probably means that Premiere is not using the Smart Render feature.

Test Results Showing Smart Render Working:Preview files set to ProRes 422 HQ, Exported to ProRes 422 HQ: 1.4 seconds.Preview files set to ProRes 422 HQ, Exported to ProRes 422: 5.6 seconds.

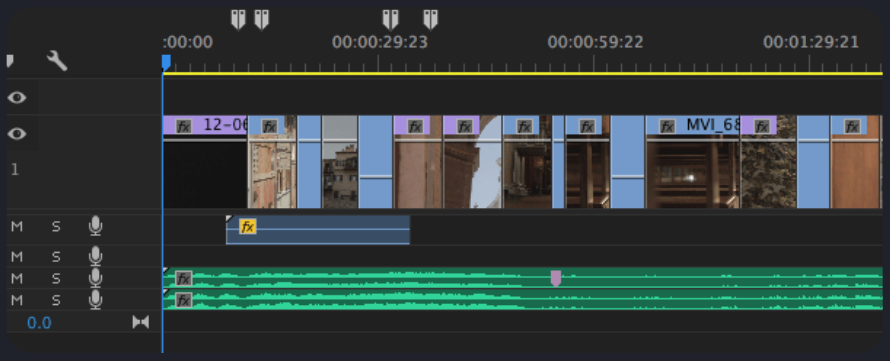
Test Results Showing Smart Render Not Working:Preview files set to ProRes 422 HQ, Exported to ProRes 422 HQ: 4.9 seconds.Preview files set to ProRes 422 HQ, Exported to ProRes 422: 5.6 seconds.

## 3. Smart Render from Source

The third trick can make a HUGE difference in export speeds, but it will only work for people who are doing only the editing in Premiere, not the color correction.

Let's imagine that you are the editor for a film, and the camera department has given you proxy files to edit with. They took the raw files from the camera and converted them into a DNxHD proxy format with a standard Rec. 709 look (which

means you don't need to add a LUT to the timeline in order to preview accurately). You are just doing the edit, so you don't need to bother with any color correction or special effects in premiere.




{caption: The grey "fx" boxes show that you have not applied any effects or color correction to these clips.}

If you export in exactly the same format as the source footage, which in this case is DNxHD, then Premiere doesn't have to render or compress at all. You haven't added any color correction or effects to the clips, so Premiere can copy the images straight from the original files, which are DNxHD, and paste them into your exported file, which is also DNxHD.

This is another way to take advantage of Premiere's Smart Rendering feature, but this time we're only using the Smart Rendering, not the Use Previews feature. When you're Smart Rendering from your source footage, you can export an entire 10-minute film in a few seconds. It's that fast.

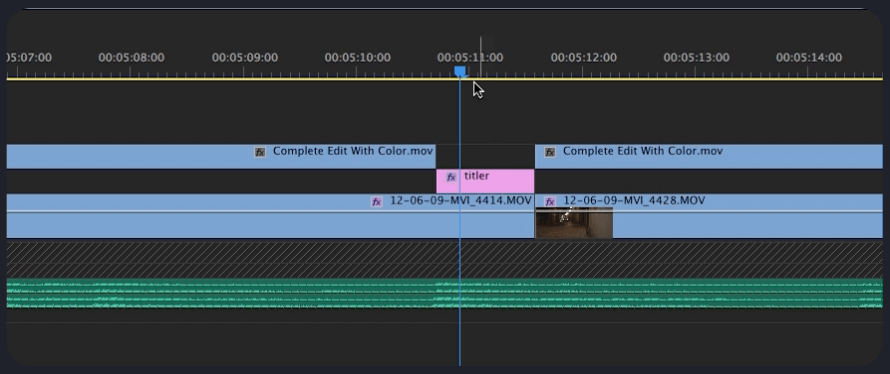
Again, since we're using the Smart Render feature, we can only use one of the codecs listed on this page. H.264 files, for instance, will not work with this workflow, but ProRes, DNxHD, and Cineform all work well.

## 4. Smart Render from Export

Trick #4 is a lot more flexible than the first three. There are no restrictions about your source footage, you don't need to pre-render the timeline before you export, and it works even if you're doing lots of effects and color correction in Premiere.

The first time that you export, you have to wait for the normal export. But if you need to make a minor change to the edit and export again, you do a smart render from your first export.

You take the first export and import it back into your project, and then lay it on top of your timeline in a new layer. Then you take the slice tool and cut away whatever portion of the timeline that's been changed for the second export.




As long as you use the same settings for both exports, Premiere will be able to copy the compressed data straight from your exported file into your second exported file. All that it has to render is what's changed, and so it exports extremely quickly.

The great thing about this method is that you don't have to prepare anything first. You didn't have to remember to pre-render your sequence. You can take advantage of this trick after the fact.

Again, this works with ProRes, DNxHD, Cineform, and all of the codecs listed on this page.

Note, in each of these 4 tricks, we care about saving time on the rendering and compression of the video not the audio. Audio is so easy to process compared to video that it's not worth bothering about. You'll notice in the image above that the audio channels are locked, and only the video portion of the export has been added back onto the timeline. The audio was not.

If the camera files are in a raw format, then this method is not a good idea, since you'll prevent the colorist from having access to the raw information. Once you've exported a raw file, you have lost some of the information inside it.

## Import into the finishing tool and check for errors

Once you've transferred the project to the colorist or finishing artist, they will import the project into their software. Depending on the type of workflow, the relinking process might happen at this stage, instead of earlier in the process.

Then the artist will also import the reference video that we made at the beginning of the conform, and place it on top of the video track with a blend mode set to Difference. The Difference blend mode will calculate the difference between the top track (the reference video) and the bottom track (the conformed video). Any pixels that are identical between the two tracks will appear black, while any pixels that are different will appear in color.

Blend Modes

Blend modes are tools for merging two images together.

Learning how to use them can be incredibly valuable for any creative professional, as they are mostly standardized across all video editing and compositing software, including Adobe Premiere, Final Cut Pro X, DaVinci Resolve, and Nuke. They are often underutilized though, as many users think of them as mysterious functions capable of only creating strange visuals. However, understanding what is objectively happening "under the hood" instead of just subjectively observing with your eyes can unlock the full potential of blend modes as powerful utilities in your workflow.

In this subsection we'll break down five blend modes: Difference, Multiply, Linear Dodge, Subtract, and Darker Color. You will learn what is actually happening in the software when you click on each one and how that knowledge can be applied to solve common problems every editor, visual effects artist, and motion graphic designer is guaranteed to encounter across his or her career.

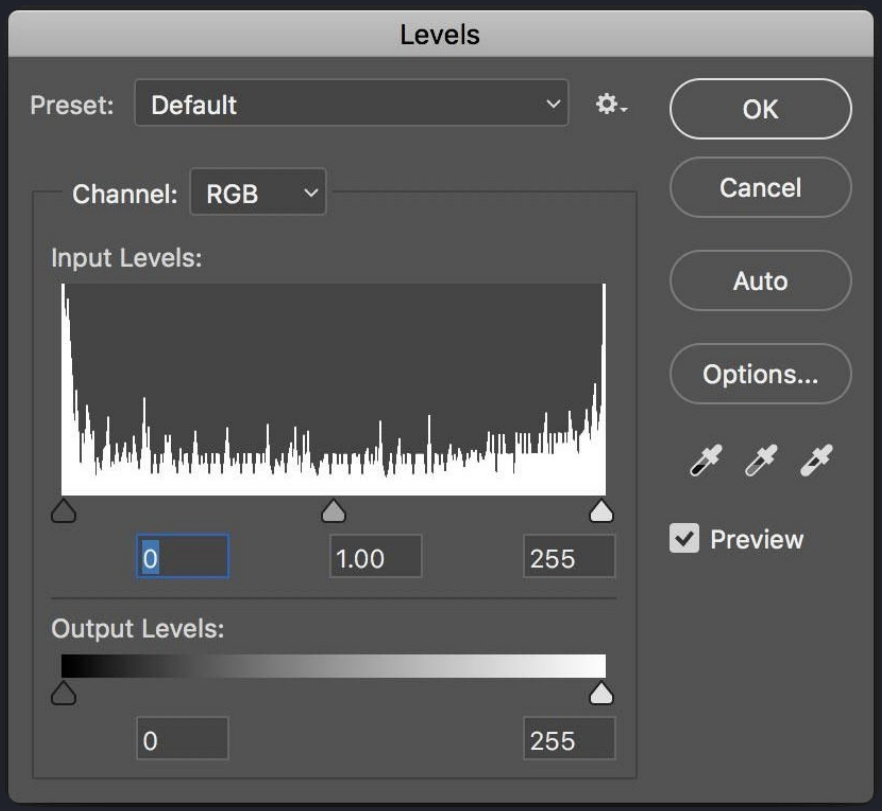
## What is a Blend Mode?

Blend modes composite two images together using a mathematical equation or algorithm that an A and B value passes through to define a new output.

Some of the less-used ones can become fairly complicated to those of us without computer science degrees, but a vast majority of blend modes are relatively simple to digest.

The "A" value is the active layer, the image you apply the blend mode to. The "B" value is the background image, the one you are blending with. Now what inputs into the A and B values?

Blend modes can access different color information, but for the purposes of the below examples, assume only the luminance value (brightness) is being used. Luminance values are typically represented ranging from 0 – 255. These numbers may look familiar to those working with Level Adjust tool in Photoshop.




When inputting into a blend mode however, these numbers are converted to a range from 0 to 1. "0" is pure black, "0.5" is middle gray and "1" is pure white. You do not need to know the exact number you are working with, as long as you understand relatively where within this range your value falls.

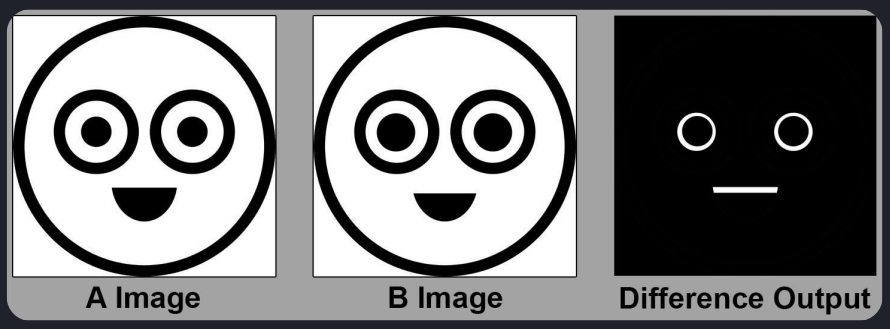
Now that we have the basic framework understood, let's get into some examples of how this could be used.

## Image Comparison with Blend Modes

Everyone has been in a situation where you need to know if two images are exactly the same (and if not, what is different about them). The aptly titled blend mode *Difference* can help you here.

The exact equation of the Difference blend mode varies slightly per software, but its essence is simple: each pixel in image A is subtracted from the corresponding pixel in image B. If the result is a negative number, it is converted into a positive number, so it actually doesn't matter whether you subtract A from B, or B from A.

If the two pixels in A and B are very similar, A – B will result in a very small number. The smaller the number, the closer to black it is, so areas in the image that are very similar appear as very dark, whereas areas of high difference are brighter.




For the purposes of simplicity, this example uses pure black (0) and pure white (1) in the image above. You can see that the identical pixels in the A and B images output black, while the pixels that are different output white. Those white parts in the blended image can then be used to identify the change in pupil size and mouth shape as the differences between the A

and B images.

Here is a real-world example. The lion is yawning, while the shot remains still, so the background is very dark, while the lion's face, which has moved between to two images, appears brighter.

Let's outline a few practical examples in which this would be helpful:

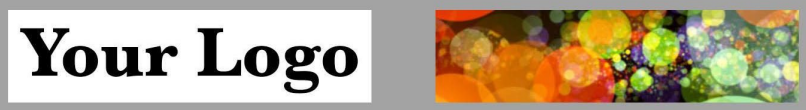
- When working with animation or visual effects, it's common to receive multiple versions of the same shot. Use the Difference blend mode to compare two versions and see exactly what has changed
- When resizing an image to match the positioning of a reference, using the Difference blend mode saves you a lot of guesswork. With the media you are scaling as input A and the reference image as input B, you can easily find the point at which your images align.
- When answering the age-old question, "Are these two

mislabeled files the same thing?" Simply line up the start points, turn on the Difference blend mode, and scrub through your timeline. If the two video files start to vary from one another, your program monitor will begin to output some very discolored images.

We can't stress enough how many varied functions the Difference blend mode can be used for. Recreate the scenarios mentioned above to start getting familiar with the function. Then keep this in the back of your head and you'll be amazed at how many times the Difference blend mode can speed up your workflow or give you absolute confidence in assessing any multitude of situations.

## Creating Masks with Blend Modes

Anyone working with graphics elements has run into this issue –a file delivered without an alpha channel. If the file is on a solid white or black background, a blend mode can save you the imperfect results of keying, the tedium of manually masking, or the agony of waiting for a graphic designer to return your call.




A Image



B Image

Various blend modes can be helpful in this situation, so let's consider the following examples:

*Multiply*: A x B. The white pixels of the active layer disappear to reveal the background layer. The black pixels remain unchanged.

*Linear Dodge (Add)*: A + B. Often thought of as the opposite of Multiply, the black pixels of the active layer disappear to reveal the background layer. The white pixels remain unchanged.

*Subtract*: B – A. The white pixels of the active layer turn black and the black pixels of the active layer disappear to reveal the background layer.

Applying what we've learned above regarding black and white inputting as 0 and 1 respectively, it is not hard to see why we get these results. In the example of Multiply, any number multiplied by 0 (black) becomes 0 and any number multiplied by 1 (white) remains the same. In the example of Linear Dodge, any value added to 1 continues to be white, as 1 is the highest value in our range. Adding 0 (black) displays the background image, as adding 0 to any value produces no change.

As with the Difference blend mode, the deeper your understanding, the more uses you will find. For example, practical muzzle flare and fire effects are often filmed in all black environments. Using the Linear Dodge to composite them easily removes the background while leaving the middle color ranges to interact with the background layer. As you experiment with these concepts, try the Screen blend mode as well. The equation, $1-(1-A)\times(1-B)$, can seem a little scary to those of us with film degrees, but if you are working with solid black and white images, the results mirror Linear Dodge.

Once you start adding other colors into the mix, you will start to see a difference in how those colors interact.

## Sky Replacements with Blend Modes

Picture this: The sun is setting behind the beautiful Los Angeles skyline. Tall buildings extend into the clouds. Nubbly palm tree silhouettes adorn the mountains extending as far as the eye can see. Now picture this: You are sitting in a dimly lit room staring at an iMac wondering how you are going to replace the sky without having to rotoscope every little palm frond. Don't worry, blend modes are here to help.

A Image

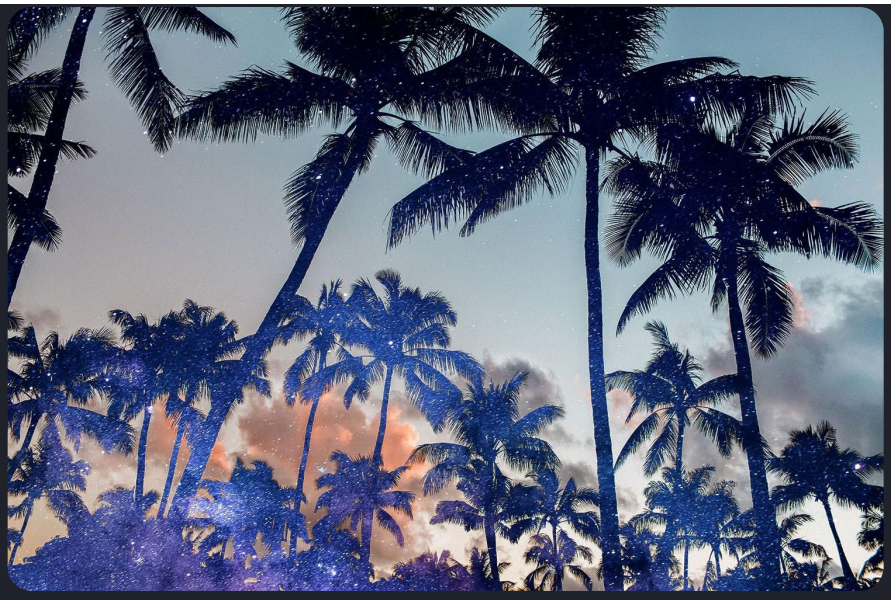
B Image

*Darker Color*: Under the hood, this blend mode is a simple algorithm that looks something like this: if (A < B), display A. If (A > B) display B. Or simply put, Darker Color compares each pixel in your active and background images, then displays whichever one is darker. This mode is slightly different than others mentioned in this article, as it does not create a new value, but chooses which of the existing values to display.

As you will see, the blown out sky is pretty well replaced by our background image. It is in no way a panacea and usually requires some additional masking, but certainly gets us to a properly composited image much faster than rotoscoping the entire foreground.

If you know what a luma key is, the Darker Color blend mode has essentially mimicked a luma key in this example, but with just one click. A chroma key (or green screen effect) would not work very well in this circumstance because they sky contains many colors.

When using Darker Color, check out its partner in crime, *Lighter Color*. It is the opposite of the Darker Color in that it displays the lighter pixels between the A and B image.



Using the same two images, the Lighter Color blend mode produced this image. It is certainly not the most practical use case, but demonstrates how, if shot properly, everyday objects can be used as masks in your post-production workflow.

The artist then watches the entire film through to confirm that nothing has gone wrong. It's unlikely that the film will be entirely black, because there will be many minor differences between the reference video and the conformed version. The conformed version will have no color correction, and no temporary effects that the editorial team added. The difference mode will highlight any serious issues, like the film being out of sync, or missing or incorrect clips being inserted.

In spite of how careful everyone is to prevent these issues, it's rare that everything just works the first time.

## The Most Common Conform Issues

Considering that the online process involves many different teams, usually in different locations, and with different software platforms, errors are always possible. Even more likely

than technical glitches are procedural errors or oversights. These can be on the part of the offline edit team, such as a misunderstanding of what the online facility needs, or could stem from the online facility not communicating their needs sufficiently.

Below are some of the most common stumbles experienced during the conform process:

## 1. Not including project files in the turn-over to an online facility

The Avid Media Composer bin (.avb) or the Adobe Premiere Project files (.prproj) are essential to the online process, as they represent the most complete "genetic" history of the project. The project file is often omitted by offline editorial teams when turning over to online facilities.

Without these files, the online facility will be unable to revert to the project timeline if something is missing in the EDL, XML, or AAF files

## 2. EDLs truncating source file names

When exporting an EDL, many of the default templates will truncate file names to only a few characters. For example, the CMX3600 EDL template will reduce original camera file names to only 8 characters. Since many modern digital cinema camera systems create camera files that are 20+ characters long, this will make conform near-impossible.

Instead, be sure to export EDLs with something like the the File32 or File129 templates in Avid Media Composer's List

Tool, or select the "32-character name" setting in Adobe Premiere Pro's EDL export window. This will keep original camera file name in tact.

## 3. Committing MultiCam Edits

If you are using Group Clips or MultiGroup clips in Avid Media Composer, all camera angles will be preserved in AAF exports unless you chooses "Commit Multicam Edits" first. Without this step AAF exports will take a disproportionately long time to export from Avid Media Composer. Additionally, the unneeded media will be included in the conform even the camera angles that did not make the final edit. For Non-Avid platforms, including multicam media may lead to misinterpreted footage, where the wrong camera angle is placed on the timeline when importing the AAF. Obviously this would resullt in conform errors, and client confusion

## 4. Stripping away CDL information from the project metadata

Some editorial processes in Avid Media Composer have been known to "strip" away CDL metadata out of the ASC_SOP and ASC_SAT columns. This can lead to confusion.

To avoid this, you should verify that an AAF still contains this metadata. Just create a fresh project in Media Composer and import the AAF, then check the CDL values to make sure they are still present.

## 5. Not including CDL Information in EDLs

This is similar to the previous issue. You can prevent it by choosing "Color Decision List" in the export settings of Media Composer's List Tool.

## 6. Providing multiple final sequences with the word "final" in it

Too often, a project file is provided to the online facility with multiple sequences entitled "final." There might be a "final version" or "final approved revised version."

To avoid confusion, it is best to provide only one sequence in a final bin or folder so there is no ambiguity as to which sequence is the final one for conform editorial

## 7. Exporting visual reference files using an H.264 codec

H.264-based Quicktimes are, paradoxically, often not friendly to common conform and grading platforms. This has to do with the interframe compression that most flavors of H.264 utilize.

Instead, use an intraframe codec, such as ProRes or DNxHD.

## 8. Renaming clips in the NLE project and losing the original camera filenames

This can be a disaster. Always be sure to copy the OCF (Original Camera File) names into another project column before renaming any clips.

## 9. Delivering “giant shovel” data dumps

Sometimes online facilities receive hard drives with literally 100% of the original camera content, without any degree of organization. This is a nightmare

Remember, online editors do not have the same degree of familiarity with the project as the offline editing team. They are going to need some help. Take the time to organize your OCFs, and maybe trim the unused takes on large project handoffs.

## 10. Transcoding unnecessarily

If you are doing the conform yourself, prior to turning over to an online facility, be wary of transcoding un-necessarily. Some editorial media types are already equal-to or better-than what the camera itself created. This makes additional transcoding redundant, time consuming, and potentially detrimental to picture quality

As an example, if you have CanonXF 50Mb 422, and you want to transcode to ProRes 422 for offline editorial, that would be a mistake. You are already editing the project with a media format that is better than what the camera created in the first place. Thus, no additional transcoding is needed during turn over.

Consult with the online facility to see what aspects of an offline editing timeline are, unexpectedly, already online-ready.

## 11. Assuming subtitles will work for the other team

For subtitles and/or closed captioning, never assume that they will appear as typed into your NLE. They will not always translate to the conform platform used by an online facility.

Rather, be sure to discuss subtitles/closed captioning with your finishing facility early in the process. Expect that, at least in some cases, subtitles will need to be completely recreated, either by the online facility or by a specialized "house" that has unique toolsets and skills to create them.

## 12. Not including LUTs or CDLs

Be ready to provide any "show LUTs" or scene LUTs or CDLs, so that the conform tool can prepare those in the color project. Conform editors often are multi-tasked into the position of preparing the content for grading and will need to apply LUTs and CDLs where needed. But if they don't have them, they will call you asking for it. It's better to just include it from the beginning.

## 13. Not preparing the conform for color

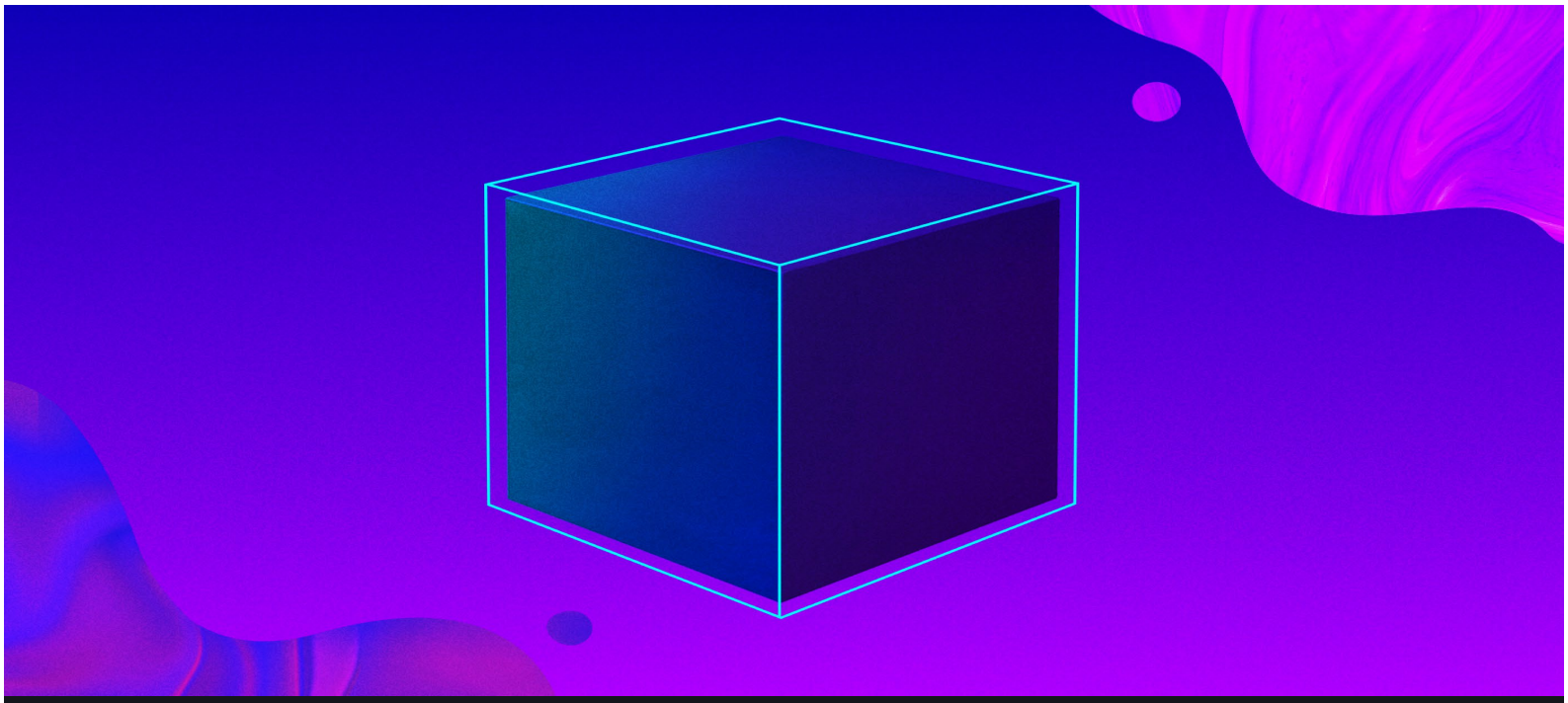
If you are turning over a flattened conform to a color-grading facility be certain to do the following:

- Remove all temp color: leaving temp color behind will interfere with a colorist's process
- Separate all titles or graphics that live on top of picture to another layer of the timeline: "baking in" the titles or graphics

over picture will make color grading difficult or impossible

- Include an A-mode EDL so that the color system can "notch" the flattened file at the edit points
- For transitions such as dissolves, be sure to include the outgoing shot and the incoming shot with sufficient handles at the end of the delivered, flattened file. This way the color grade can be applied to the handles of the outgoing and incoming shots, and the dissolve can be reapplied after the grade is complete
- For composites done in the NLE, be sure to include the V1 layer and the V2 layer (and as many layers are needed) at the end of the delivered, flattened file; the layers will need to be graded independently, and then re-assembled in the conform tool after the grades are applied.
- Be sure to indicate to the online facility what fonts were used for titles, if titles are to be re-created during the online
- For stills, include the highest resolution still available for grading when possible (even if it isn't used)

Contributed by Matt Schneider, Director of Technology, Technicolor New York.

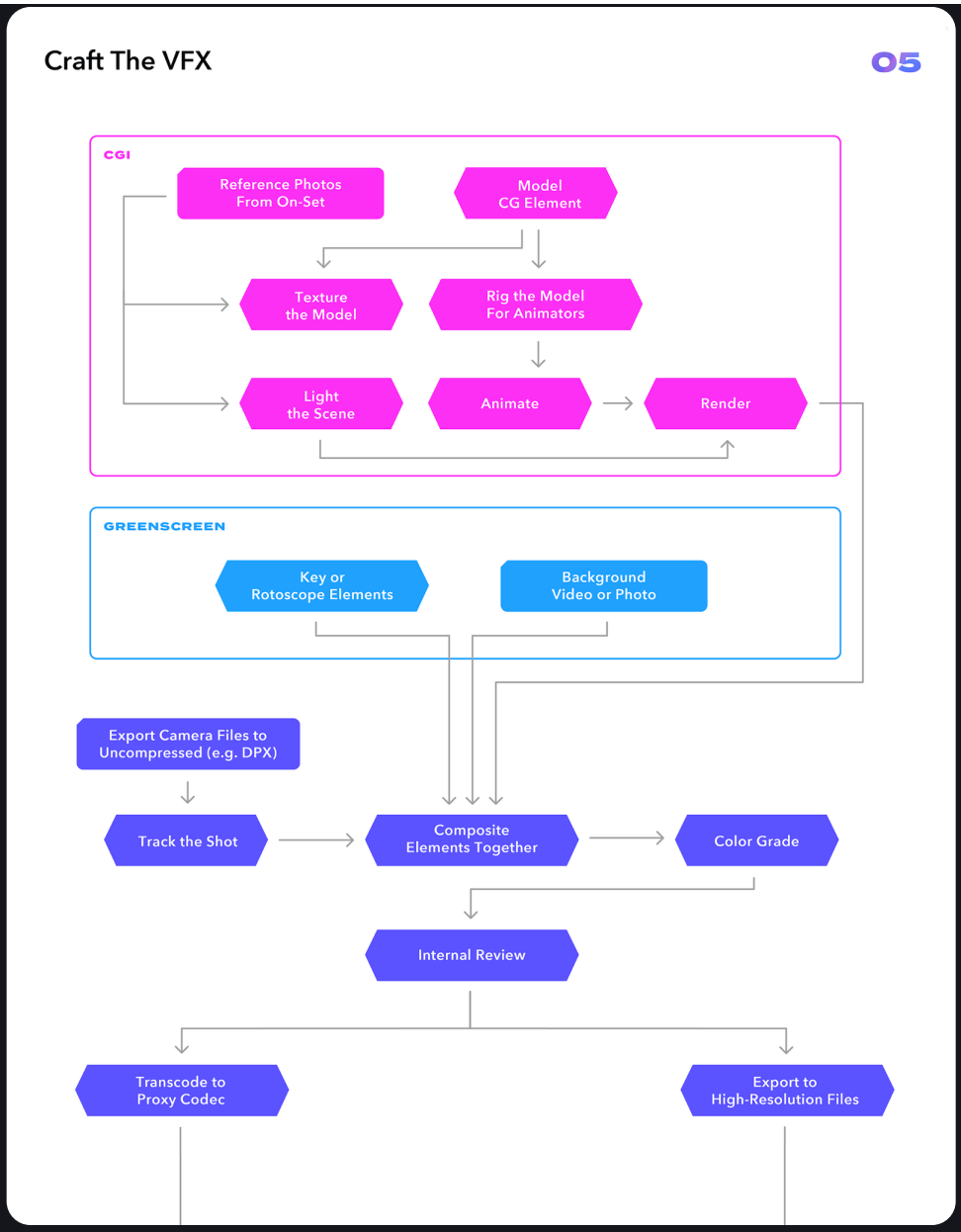


CHAPTER 05

# VFX

We all know that VFX creates new worlds, but it can also save a huge amount of time and money in this world (when done properly).

*12 minute read*




When you hear visual effects (VFX), you probably first think of monsters and spaceships. You're not wrong, but as the tools have gotten cheaper and easier to use, VFX have become ubiquitous in motion pictures. Even a simple drama or corporate video project frequently makes use of VFX to fix small errors or save the production time and money.

## Special Effects vs. Visual Effects

Special Effects, Visual Effects, Virtual Effects, Digital Effects, Computer Effects, and even just Effects. You have definitely

heard these terms used interchangeably at some point in your career. But they do not all mean the same thing.

While the average cinema-goer could be forgiven for mixing up these industry-specific terms, many video professionals still occasionally conflate them. You really should understand the specifics of this terminology for your project, so as to avoid any miscommunication or confusion among your post-production team.

## Special Effects (SFX)

Special Effects are illusions, simulations, or mechanisms within a shot that are based in the "real" world. They are often called *practical effects*, and include things like pyrotechnics, animatronic puppets, car crashes, prosthetics, and miniature models. These effects are recorded by the camera with the rest of the shot on set, rather than being added in post.

SFX have been around since almost the very beginning of cinema. Nearly a century of technological and artistic achievement have made it possible to portray incredible things on screen, such as breathtaking space travel sequences (*2001: A Space Odyssey*, 1968), realistic dinosaurs (*Jurassic Park*, 1993), and time-warping sets (Inception, 2010), all created practically.

## Visual effects (VFX)

Visual Effects, on the other hand, are added after the footage has been recorded, usually with a computer. Most often, VFX are elements integrated into a live action shot. These elements can add something that was not there before (such as a new background over a green screen) or remove

something that was (like an unwanted logo). Some VFX shots are entirely made up of post-production elements, with only small live-action elements added in (like CGI scenes with faces composited onto models). To clarify the terminology, VFX encompass the terms digital effects, computer effects, and virtual effects, which sometimes refer to specialities in the field.

VFX developed as the capabilities of computers and digital systems began to grow. A notable early example was *Star Wars* (1977), which used practical effects like miniature models spaceships and explosions, but composited these live-action elements together with extra virtual details and effects. As computer systems have matured, it's now possible for incredibly complex scenes to be created, from scratch, entirely in the digital world.

## SFX vs VFX

SFX and VFX work to do the same thing–to make things look real that aren't. Both are just tools, and each can be used to produce stunning results. Now more than ever, filmmakers have incredible options for creating masterful effects on screen.

When deciding how to achieve the effects you want for your project, the chief considerations between SFX and VFX are the tradeoffs.

SFX require tremendous precision and on-set planning to pull off successfully. However, this can deliver unparalleled visual realism, because they are being captured in frame by real camera.

VFX still require precision and planning, but they allow you to

do more after the fact. Also, some things just aren't possible with practical effects, and computer graphics have opened up a whole new world of effects for even low-budget projects. That said, VFX can introduce new problems for a production. Actors might have a difficult time reacting to imaginary monsters/explosions/etc, so even if a visual effect looks good, the rest of the scene might be poor because of it.

The final considerations for choosing SFX and VFX for your project are logistical. Depending on the exact effect you want to achieve, it could be appropriate to use either a practical or digital approach. It all comes down to what you want, when you want it, and how much you're willing to pay. Different circumstances will call for different approaches.

At the end of the day, it all comes down to what serves your project's needs better.

## Invisible Effects

The vast majority of VFX that appear in modern films are "invisible effects," which means that (ideally) the audience has no idea they're watching something that's either been digitally altered or is completely synthetic.

Perhaps you have a scene in which a house catches fire. You could build a dummy shell of a house and actually set it on fire, or you could save a production time and money by adding computer-generated (CG) fire.

The fire in "Get Out" was computer-generated.

Or perhaps you recorded a killer CEO interview for your latest marketing video, but didn't realize that the CEO was wearing a shirt with the Ralph Lauren Polo symbol. Since you're probably not excited about contacting Ralph Lauren's PR team, you're going to need to use VFX to remove it.

Or maybe you're shooting a period piece and want to remove modern "tells" such as signage or streetlights. Or you have an actor who has two perfectly functional legs but is supposed to look like a double amputee.

Suffice it to say, since digital visual effects applications are now widely available, it's more likely than not that any feature film (or high-end commercial or mainstream TV show) will use them to fix something that the viewer will never suspect as being inauthentic.

Here's a fun question: What was the last Best Picture winner that didn't use VFX? Hint: it was more than 20 years ago.

## Planning For VFX

The initial stage of VFX begins in pre-production. Major VFX require

planning and coordination between production and post-production crews, so that every technical requirement of the shot is planned for and properly executed.

On a smaller project, the VFX Editor is the liaison between the Editorial and VFX teams. To fill this role you must have an understanding of both worlds and be able to keep meticulous track of each shot.

On a VFX-intensive film, there may be thousands of effects shots distributed among numerous vendors. In such cases, a post-production supervisor (who may also have one or more assistants or coordinators working with them) functions as the liaison between the editorial team and all the VFX vendors. It's an important job, requiring fluency in the language of visual effects, superior attention to detail, and a cool head to deal with both logistical and creative detours.

## Editorial and VFX

If you're editing a project that incorporates VFX, you'll need to use some form of placeholder during the offline. If the project you're working on is adding visual effects to a live action plate, the plate itself will act as the placeholder. Or, if the foreground element is, for example, a live actor performing over green screen, the performance acts as the placeholder.

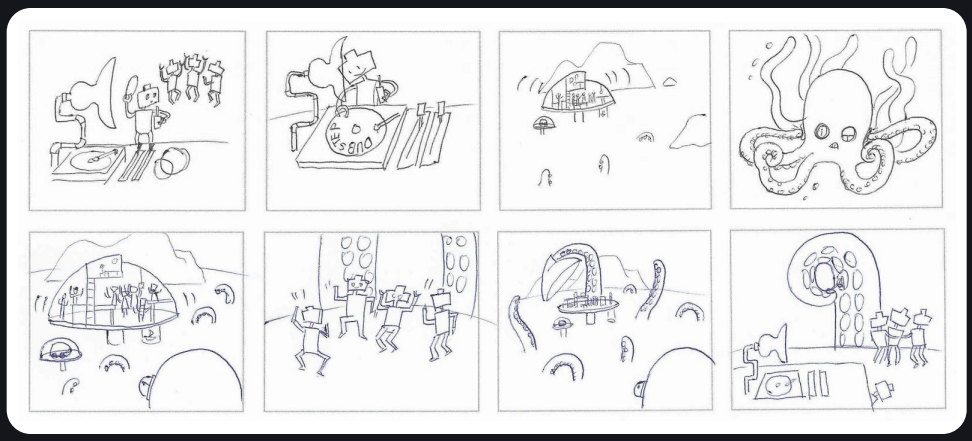
If, however, you're editing a piece with completely computer-generated shots or sequences, you'll need to create a placeholder.

- Text-o-matic – The most basic type of placeholder, they're simply title cards with shot descriptions. "VFX – Spaceship flies by camera," for example.
- Storyboards – typically hand-drawn still images showing the framing and intended action of certain shorts that provide the editor with a

pre-timed space in which to drop the completed (or work-in-progress) VFX and assist in choosing the best surrounding live action shots.

- Animatics – A group of storyboards laid out on a timeline to give a sense of pace and timing. Helpful in lengthier sequences, they allow the editor to work with music or voice-over to help create the flow of the sequence(s) prior to commencing computer animation.

- Pre-visualization (or Previz) – Wireframe or rough 3D animations that function as stand-ins for sequences designed to display motion and timing; they're especially important for action sequences.



Storyboard by agency Critical Mass in collaboration with the production company Coat of Arms on Google brand video.

## Handing off to VFX

"John Wick 2" editor Evan Schiff working with animatics.

In the case of shots or sequences that combine live action with synthetic elements, the editorial team is responsible for selecting the takes that will be supplied to the VFX house before they can begin work. Usually, someone from the editorial team (probably an assistant editor or VFX editor) will create a rough temporary version of the shot in the NLE. This allows the editor to make sure that the performance works in the scene before sending the shot off, or that the background is the best one in which to integrate the effects. You really don't want to have the VFX team start on a shot only to realize that it might have worked better had you chosen a different take.

Once the editor is confident of the shot's placement in the sequence, the VFX editor (or an assistant editor) will prep several files.

First, the VFX editor will identify the original camera file(s) that corresponds to the shot to be worked on because the VFX team needs the full resolution elements to ensure the highest quality. In the most complex visual effects scenarios, there may be multiple backgrounds, foreground elements, mechanical rigs over which characters will be placed, or lighting passes on the background. The VFX house will need the high-resolution assets for everything that is to be incorporated into the shot and will use some sort of asset

management tracking to ensure that all the elements are accounted for—it's not unusual for VFX shots to incorporate dozens of elements.

You can either supply the entire original source file (which may be much longer than the section that's actually used in the sequence), or you can export a smaller sub-clip. If you send a sub-clip, make sure that you include handles and use a proper Mezzanine codec. It's also handy to send a copy of the rough cut sequence so the VFX vendor understands the context into which the shots they're working on will fit.

As always, proper file naming and metadata management are vital to this process, so that everyone can keep track of everything that's been sent off and where exactly in the sequence they belong. The VFX editor (and/or the post-production supervisor/coordinator) and the VFX house will each use detailed lists and spreadsheets (or some sort of off-the-shelf or custom project management software) to track the status of each shot throughout the process.

When a VFX shot is completed and signed off on, the VFX house will render out the finished version of the shot to a high-quality Mezzanine codec or to an uncompressed format and send it back to the editorial team.

The VFX editor will then create a proxy with the same codec that's being used for the rest of editorial and drop it back into the sequence to make sure that it works as planned.

The file sent from the VFX team is treated like a new camera original file and should be backed up and stored along with the rest of the camera files so that it can be easily found when it's time to link the sequence back to them (aka conform).

# Compositing

Compositing is the task of combining multiple images together so that they appear to be part of the same shot. While technically it's the final step in the VFX pipeline, it's also the most fundamental.

Image © Disney. All rights reserved.

Compositing is a precise endeavor because the two elements (or, with a complex shot, the dozens of elements) were created separately. In order to convince the audience that Iron Man and Pepper were both actually in the room at the same time, the compositor may have to spend hours finessing details such as lighting, focus, shadows, reflections, and atmospherics in order to make the various elements look as though they all inhabit the same space. The human eye is surprisingly good at spotting the deception, particularly in instances where we know what a human (or animal or airplane) looks like in real life, in its real environment. A poorly executed composite can easily result in the viewer feeling pulled out of the action or the story.

Compositing

Coming Soon!

## Mattes

In essence, a holdout matte is just what it sounds like—a cutout of a foreground object that allows you to isolate it from its background and place it over a new background.

In order to composite multiple foreground (or mid-ground) elements over a background in compositing, you need a matte for each element. There are many ways to create a matte. Usually, if you are adding a CG element to a live-action shot, the VFX software can generate a matte automatically. If you need to cut out a piece of a live-action shot, you can cut them out by hand (aka rotoscoping) or use a green screen or blue screen (aka keying).

Mattes

Mattes are the foundation of compositing, allowing you to combine multiple images together. In essence, a mat is like a cut out – it tells the computer which areas of the image should be visible and which should be transparent, allowing other images to show through. In a green screen shot, a matte is created by removing all of the green from the image.

Once a matte is generated, you can enhance, repair, or create entirely new shots by combining it with other mattes in the foreground and background.

You can sift mattes into two categories: static and traveling.

## Static Mattes

If you want to isolate a shape in your shot that doesn't change over time, you create a static matte. Why?

Think of a wide establishing shot of a house executed from a camera on a tripod. The lawn has a garden gnome. The gnome wasn't supposed to be in the shot. To remove it, you could create a matte to separate it from the rest of the shot, then use compositing techniques coupled with your software of choice to do so. Since the shape of the gnome doesn't change over time, this matte would be static.

## Traveling Mattes

If you're isolating a shape that change shape over time, you create a traveling matte. Why?

Think of a hero leaping from one skyscraper to another. The actor is running, jumping, flailing in the air. If the actor's performance is shot in a stage lit and designed for visual effects work, you can generate a matte by removing the existing background to isolate the actor's performance, which can be combined with shots of the desired environment. Since the shape of the actor (definitely) changes over time, this matte would be traveling.

Matte creation has evolved over the years. When computer-generated imaging became a visual effects fixture, mattes took the form of transparency information in digital images.

## Alpha Channels

Digital images typically contain three channels of information to represent color: red, green, and blue. Those three colors are combined to make the final image. But some images have a fourth channel, an alpha channel. Alpha channels contain transparency information that tell your image compositing software which parts of the image are visible, and which are transparent. In other words, mattes can be generated in one piece of software and then bundled with your image to be sent off to another piece of software.

Here are some common methods for generating mattes in computer-generated images, still and motion, with alpha channels.

## Luma Key

Luma keys operate on the brightness levels, or luminance values, of an image. A matte can be generated by applying a luma key and designating which brightness levels should be transparent and which should be opaque. This is known as pulling a key.

## Chroma Key

Chroma keys let you select a specific color in an image (i.e. the chrominance value), designating anything with that color transparent. This is also known as pulling a key. Performances shot in green or blue physical environments were executed with chroma keying in mind.

### Masks

Transparency can also be defined using masks, which is a shape is drawn over a portion of a shot. You can then define which area in relation to the shape will be transparent, inside or outside. You might think of masks for generating static mattes only, but traveling mattes can also be generated by moving mask points over time as the shape of your masked element changes. Rotoscoping is the process of using masks to create mattes.

### Transparency

A matte defines which areas of an image should be transparent, but a matte can also have many levels of semi-transparency. It's very common for an artist to fade the edge of a matte so that it blends in smoothly with the background image, for instance.

## Common Applications of Mattes

Knowing what a matte is and how it can be defined in software, let's look at some common applications of mattes in filmmaking.

### Matte Paintings

Matte paintings can add visual elements in the frame, extend existing sets, or serve as the set itself. For a long time, matte paintings were actual paintings on glass.

https://www.youtube.com/watch?v=kPcEFHA3X0c

https://twitter.com/tvaziri/status/1060605631646720001

Although the medium changed, the principles haven't. Composition, perspective, and lighting are all carefully considered to execute a convincing matte painting.

For years, matte paintings were mostly static. But with software and increasingly sophisticated techniques, many are now traveling mattes, allowing filmmakers to replace just about anything, foreground or background, in a shot.

## Garbage Mattes

Most visual effects shots start out with 'something in the shot'. Could be lighting gear. Could be some other rigging. Could be a crew member! Usually, all of these are in the shot out of necessity.

To hide these, a garbage matte can be generated. The intent is to keep the wanted elements visible in the shot while hiding the unwanted bits (i.e. the garbage). Depending on how complex the desirable elements' shape changes over time, it could be a static or traveling matte. In software, these usually take the form of shape masks.

## Holdout Mattes

Holdout mattes are connected with keying. When you're pulling a key, you might discover your subject was reflecting a

bit more of your chroma key background than imagined. Portions of the body might disappear despite your best keying efforts. Using a holdout matte, you can duplicate a piece of footage in your software, draw over the portion you want to re-include, and remove the keyer applied. Again, depending on how the desirable elements' shape changes over time, it could be a static or traveling matte. These are also usually in the form of shape masks.

No matter how sophisticated or layered a shot might be, it all comes back to the matte. Knowing the difference between a static and traveling matte, how they can be applied, and how you might wield them in service to the story will only make your story better. It will also make you a valued team member and provide even more value to the client.

Contributed by Isaac Terronez, Producer, video editor, and technologist.

## Keying

Keying is based on shooting a foreground subject against a color that is easily isolated and removed digitally, thus creating a holdout matte for the foreground image that allows it to be placed over a different background.

Achieving a usable key requires several carefully executed steps, which begin on set.

Ingenuity Studios VFX reel. All rights reserved.

It's also important to light the green screen evenly across the entire surface, so there are no shadows or hot spots. And you will want to make sure there's adequate separation between the green screen and the actor or object, so that green "spill" doesn't bounce onto it. A poorly-lit green screen will yield an incomplete key resulting in a faulty matte that will likely need to be fixed by hand later—which defeats the purpose of using the green screen in the first place.

It's also vital that nothing else in your scene, especially nothing that your actors are wearing, is the same color as the screen. If your actor is wearing a green shirt in front of green screen, their shirt will disappear along with the background. The reason why bright green and blue are used is because those colors almost never appear in skin tones, so you don't have to worry about parts of your actors vanishing (unless they have colored hair!). It's also worth noting that depending on the quality of the software you're using to key out the background, very fine details can disappear, too (such as wisps of hair or feathers, which you may want to keep in mind while casting

or costuming).

Whatever keying software you use, you'll select the color you wish to key out from the shot. This may require some refinement, as the exact hue of the background will vary depending on lighting and shadows. Once you isolate the color range, remove it, and if you've shot and lit your scene correctly, your matte is complete, ready to be composited together with your background image. In many cases, however, mattes may need some fine tuning in order to neither "clamp" the image (giving it a hard-edged and slightly unnatural appearance when placed over the new background), nor to leave the image with too much of a green border, which can result in visible matte lines in the final composite.

Keying

Keying is the process of separating and isolating elements of an image by their color or brightness. It's often done for visual effects (such as to remove green screens), or in color correction (to add warmth just to skin tones).

It begins with selecting a color or brightness value in a shot, which will be rendered transparent. This is called *pulling a key*. If a shot is planned and executed correctly, pulling a key may be straightforward, but it usually requires a bit of finesse.

There are two primary types of keys: luma keys and chroma keys.

Luma keys operate on the brightness levels, or luminance values, of an image. You can generate a matte by applying a luma key to your shot, then designating which brightness level is transparent and which is opaque.

Chroma keys let you select a specific color in an image (i.e. the chrominance value), designating anything with that color

transparent. Performances shot in green or blue physical environments were executed with chroma keying in mind.

## Why Green?

Green is a color that isn't commonly found in, or on, humans. In instances where green is an integral part of a character's wardrobe or costume, shooting on blue is the next choice.

From a camera standpoint, most digital camera sensors use a Bayer Pattern.

Bayer Pattern sensors have twice as many green photosites than red or blue. More photosites means more information to work with in the captured image, so green is the default choice for digital cameras.

Here's a 3-step approach to keying shots:

1. Outer
2. Inner (Core)
3. Edge

## Outer Key

Once you apply a chroma key, focus on cleaning up the outer portions of the shot. Start by selecting the main hue of green that will become your transparent background. Then refine the key, perhaps with some Grow or Shrink controls, to encompass all the detail you want to keep. Even if it's hair blowing, make sure you get it all. You'll want a wider band around the subject you're keying.

Shooting for the keying process is rarely perfect. Uneven light

levels, shadows, wrinkles, seams, other objects or people in the shot – all of these exist, even on big-budget feature films.

The solution? Before you pull the key, create a garbage matte. Garbage mattes are quickly drawn masks that eliminate larger, unwanted elements in the shot.

### Inner (Core) Key

Focus on shrinking this inner, or core, band. This may eat into your image for now, but that's OK. Again, refine the key, perhaps with some Shrink controls, encompassing all the detail you want to keep. That green band should be gone, with a bit of loss around the edge of the image.

### Edge

Now it's time to finesse the edge. Feather and erode controls, tuning black levels and white levels, and edge color correction will restore any imaging lost.

If a series of shots were performed on the same day under the same conditions with the same camera, you could save a preset from one shot to reduce work on the rest of the shots. More than likely, though, each shot will have its own quirks to address. Have the mindset: a preset's a good start, but more work is likely to get it right.

## "This is a hard key to pull."

Like any process, there's bound to be trouble along the way. Here are some common problems you'll discover while keying.

## Spill Suppression

Spill occurs when the color of the background is reflected on the subject. Depending on how much color spills onto the subject, unwanted image loss may occur while keying the shot. Remember that a green-screen chroma key works by removing everything that's green. If the green background is reflecting off of your subject's sweaty arm, then the chroma is going to remove part of their arm as well.

Thankfully, many keyers come with some controllable form of spill suppression. This can be addressed during the Inner (Core) stage.

Also, applying a holdout matte can restore any unwanted image loss from tuning spill suppression. Or a holdout matte alone may solve the problem.

## Color Correction

Your subject's color may shift during the course of keying, so you may need to compensate for that afterward. Also, you might perform color correction to match the brightness levels or lighting angles of other shots you'll composite. Time and budget constraints will dictate how much can or will be corrected.

## Multiple Keys & Corrections

It's also possible that a shot will require multiple keys and corrections before it's released. It's sometimes necessary to

layer several different keys together, especially if the greenscreen wasn't evenly lit.

## Rotoscope

When all else fails, it's sometimes necessary to resort to rotoscoping (manually cutting out the subject from the background) if they key really doesn't cooperate. Sometimes, in a tricky shot, the left of an actor's body may separate nicely from the greenscreen with a good chroma key, but the right side of the key may fail. In those cases, it's common to accept the key where it works and then use roto to fix the areas where it doesn't.

Contributed by Isaac Terronez, Producer, video editor, and technologist.

## Rotoscoping

What happens if your matte isn't perfect just by virtue of keying? Rotoscoping (or roto) is the frame-by-frame process of tracing an object as it moves through a shot to create a matte so that it can be combined with other images. This is usually done manually or semi-manually, and can be extremely time-consuming. But it is, in many cases, the only solution.

In order to rotoscope an object, the artist must trace its outline perfectly, by hand, and then adjust the tracing for every single frame. Since most films are captured at 24 frames per second, one minute of rotoscoped footage may require the artist to adjust the tracing for 1,440 individual frames.

Some advanced tools can take a single rotoscoped frame, examine the footage, and adjust the rotoscope automatically to fit the character's movement, but even the best tracking tools often fail. In the best case, the artist only needs to make adjustments every several frames, but it's quite common for an artist to have to touch every single frame.

Rotoscoping is also used as an animation technique, in which the animator draws over a live-action character's movements frame by frame to create the look of moving art.

Rotoscoping

Rotoscoping is another technique for generating a traveling matte. In practice, it's known as being one of the most time- and labor-intensive methods of doing so. So why is it still in use?

## Art

Sometimes rotoscoping is the only way a filmmaker can achieve a particular visual grammar. For years, animators used rotoscoping to capture specific types of real-world movement that couldn't be effectively captured any other way. More recently, filmmakers have used rotoscoping as a live-action augmentation tool, producing images that take the audience beyond reality. Even the surreal.

Although having its roots in the arts, rotoscoping is used for more commonly for practical applications. Let's look at some.

## Practical Uses

## Removing the Unwanted

Let's say you have a shot with some unwanted elements in the frame. You *could* draw masks around the regions to remove if:

- The camera's in a fixed position
- The unwanted items don't move
- The onscreen talent never intersects with any of these yet-to-be-removed bits,

Chances are, none of these will be true.

To remove the unwanted, you'll create a *garbage matte*. Garbage mattes hide the unwanted (i.e. garbage) in the frame. They're meant to be quickly drawn over regions for removal, then move on. But what if the shape of the unwanted is complex? Or finely-detailed? What if it intersects with something you want to keep in the frame? This is where rotoscoping comes in.

Starting with a frame with the most detail, trace around the unwanted item(s) with a series of points to create a mask hiding the unwanted elements. Frame-by-frame, forwards and backwards in time, you'll change the position of different points on the mask to match the shape of the unwanted piece. On playback, you'll refine and adjust these points until the garbage is removed.

## Restoring Portions of Shots

With shots intended for keying, you'll frequently confront *color spill* on a subject. Also, someone may have worn clothing that matches the color of the background for removal. Once a chroma key's applied, you'll see unwanted image loss in the

form of splotchy patterns or gaping holes. When key adjustment and refinement isn't cutting it, how can you bring back these lost portions of your shot?

You'll create a *holdout matte*. In most compositing environments, holdout matte creation involves duplicating the keyed shot as a new layer, removing the key from the top layer, creating a mask as a series of points around the item to restore, then changing the positions of the points on the mask as needed.

Like garbage mattes, the precision of a holdout matte's shape will depend on the area to be restored. Additional color correction may be required if the source shot was chroma keyed.

## Combined With Motion Tracking

Yes, rotoscoping is traditionally time- and labor-intensive. But with improvements in motion tracking software, rotoscoping can be *less* taxing.

Rotoscoping with motion tracking begins with a mask drawn with a series of points at one frame. Unlike manual rotoscoping, motion tracking software can determine changes in a mask's shape over time. It does this by looking at the source footage in relation to the mask's shape. Instead of manually changing mask point positions frame-by-frame, you can jump ahead or behind *in chunks of time*, change those point positions, and the software can determine the motion in between. Results can vary based on the complexity of an object's shape and/or a shot's visual fidelity. But, with motion tracking, rotoscoping effort can be significantly reduced.

# CG (or CGI)

Computer Graphics or Computer Generated Imaging encompasses any kind of images that are generated by a computer instead of by a camera. These are often fantastical creatures or objects, but they can also be perfectly normal things used in invisible effects. If your character is doing something extremely dangerous, for instance, you may actually use a completely CGI version of the character, called a “digital double.”

Ingenuity Studios VFX reel. All rights reserved.

The basic process for producing a 3D character or object starts with modeling, which involves creating a detailed rigid 3D sculpture

which will subsequently be "skinned" or painted. Rigging involves taking that model and assigning it controls and points that will enable it to move according to the animator's needs. Once the model is rigged, the animator takes over, and creates its movements. After that, textures and shaders (that simulate such things as fur or hair) are applied. Lighting the CG character comes next, so that it looks as though it fits into its environment and believably interacts with the other objects, characters, or backgrounds. Finally, the CG is rendered, producing the final element and matte so that the compositor can do the fine-tuning to integrate all of the elements into the final shot.

Does all of that sound simple? It isn't. Creating a believable CG character or scene that looks like it was photographed is a huge challenge involving numerous complex tasks that may require anywhere from several to several hundred skilled artists to complete.

In the case of the burning house that we mentioned earlier, it could be possible to create that shot with or without CG. You could shoot video of fire against a black background and then composite it into a shot of the house. In that case, you're combining two live-action shots, so there is no CG involved. On the other hand, you could simulate the fire using computer-generated flames. Simulating natural elements like fire and water take some know-how and finesse (even with the amazing off-the-shelf software nowadays that generates natural phenomena), but the advantage is that you may be able to place them more precisely—and no one is harmed in the process.

CG

Not *too* long ago, if a character, vehicle, or environment made the leap from artist's imagination to the real world, it was planned, sculpted, painted, posed, lit, and shot on film. Filmmakers have partnered with technologists to augment, and in many instances replace, the traditional art disciplines in

the pipeline with Computer Graphics (CG), or Computer Generated Imaging (CGI). How?

Let's examine at the anatomy of a computer-generated asset.

## Modeling

In the modeling phase, a shape, or combination of shapes, is transformed into something recognizable. Instead of molding and carving clay, plastic, or resin, polygons are defined (then refined) in software. Lines and facets are divided and re-divided to form smooth curves and surfaces. Shapes are combined to add, or take away, mass from an object. With the right time, budget, and skill, the proverbial block of marble can become the most necessary version of David.

## Lighting

You can also create light sources that mimic real lights in our physical world. Properties such as intensity, color temperature, and fall off are available and configurable. Lights that serve as key, fill, rim, bounce, and kick lights can be placed anywhere you want. Create as many lights as your shot calls for. The rendering engine, a specialized software package that generates your image, may provide additional features for shaping and refining the lights in your scene.

## Cameras

Virtual cameras can be generated and placed anywhere in the scene. Unlike a physical set, you can also create as many as

you need, each with different focal length or lens emulation options. Need a dolly shot? A crane shot? A handheld? You can animate the motion of any of these cameras, perhaps placing all three in the same scene without worrying about tripping over dolly track or electrical.

## Rigging

If modeling creates the body of an object, rigging creates the skeleton. Rigs change a model's shape at different points of articulation. This is called deformation. You can specify how a rig deforms a particular area in a model, such as the range of motion in an elbow or knee.

## Shaders

What color is this thing? Is it shiny, or dull? Transparent? Translucent?

How does it feel? Rough? Smooth? Does it have a repeating pattern? A random one? A bit of both?

Questions like these are answered with shaders. Color, texture, and specular highlights are just a sampler of some of the attributes you can define when designing and applying shaders to a model.

## Rendering

Rendering takes all of this data – the modeling, lighting, cameras, rigging, and shaders – and generates an image. It

may also generate a matte (alpha channel) for compositing.

Rendering engines are specialized pieces of software that create these images in their own ways. Some are known for more photorealistic output. Others for style. Others for ease of use. All of them are known for one thing: time consumption. Being aware of you and your teams hardware and software capabilities will guide you when scheduling time to create the shots that will eventually replace any storyboards in your timeline.

Contributed by Isaac Terronez, Producer, video editor, and technologist.

## Motion Tracking

Motion tracking (also called Match Moving) allows you to insert an element (either CG or live-action) into a moving live-action shot. In our earlier example, we used CG to make it look like a house was on fire. If the camera is locked off, you simply need to position the fire elements in the right spot and composite. But if the camera is moving, the fire elements will need to be tracked to the house to match the house's position throughout the camera move.

Motion tracking involves translating the camera's movement in the live action scene and assigning that same movement to a virtual camera in the tracking software so that the element to be inserted matches the speed, angle, and scale of the background plate.

This technique is commonly used for screen replacement within scenes. You might see a TV or computer screen with plus signs or circles in the corners, and alternating color quadrants dotting the background, to aid in the collection of tracking data on the set. The visual effects crew will generally take measurements throughout the

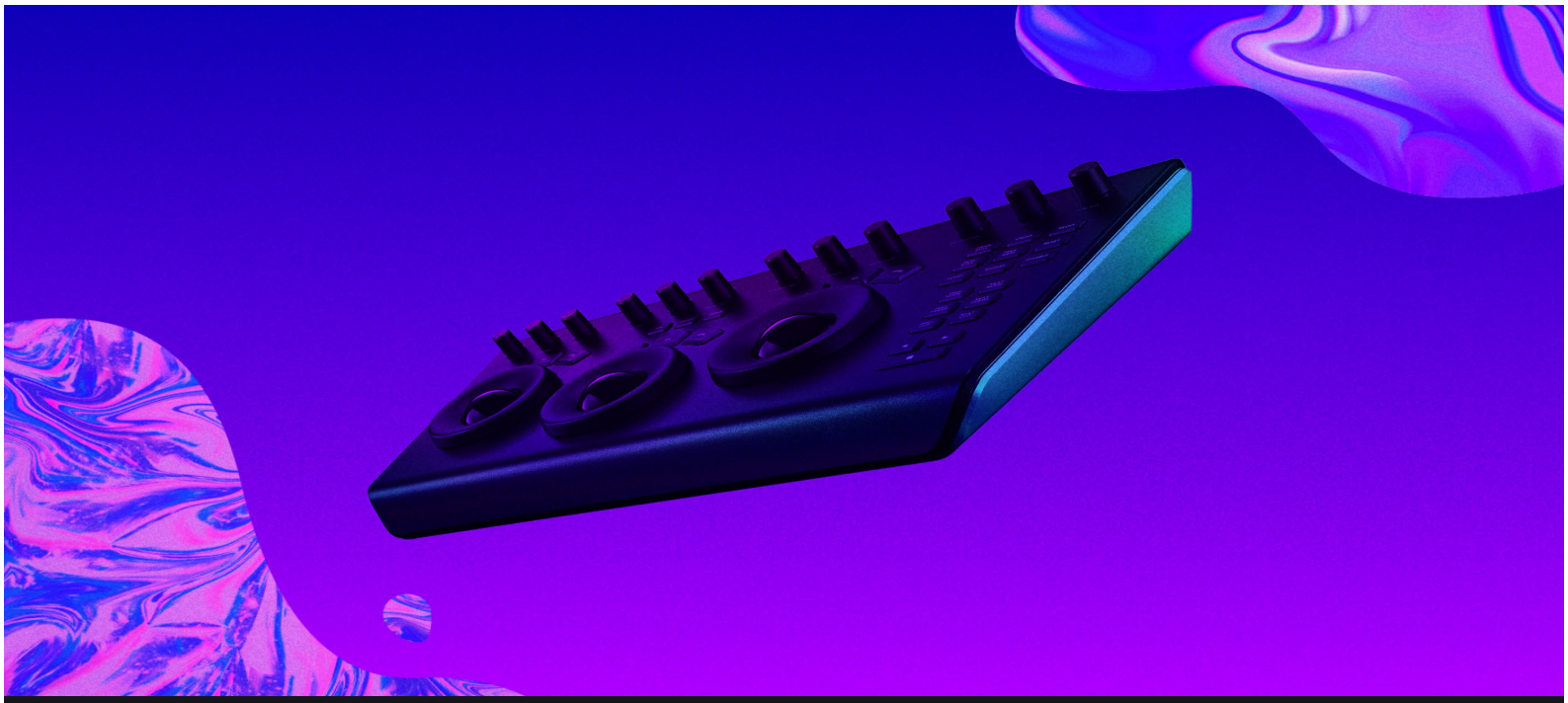
camera move either using digital surveying equipment or LIDAR (laser imaging detection and ranging). In the most complicated instances (for example, "The Equalizer 2" hurricane sequence), they needed to rebuild the whole town in the computer so they could hold out the actors and buildings throughout the camera moves and composite them back into the sequence along with all the CG atmospheric elements that were behind and in front of them.

Once the data has been collected and fed into the software, the software can track the changing position and size of the reference points over the course of the move. With that data, the screen can be replaced in post, freeing up the production crew from worries about reflections, light issues, or image flicker that are common when recording screens.

Because small, high-quality cameras and motorized gimbal technology have made it easier than ever to move the camera, motion tracking has become increasingly common in VFX shots.

Motion Tracking

Coming Soon!

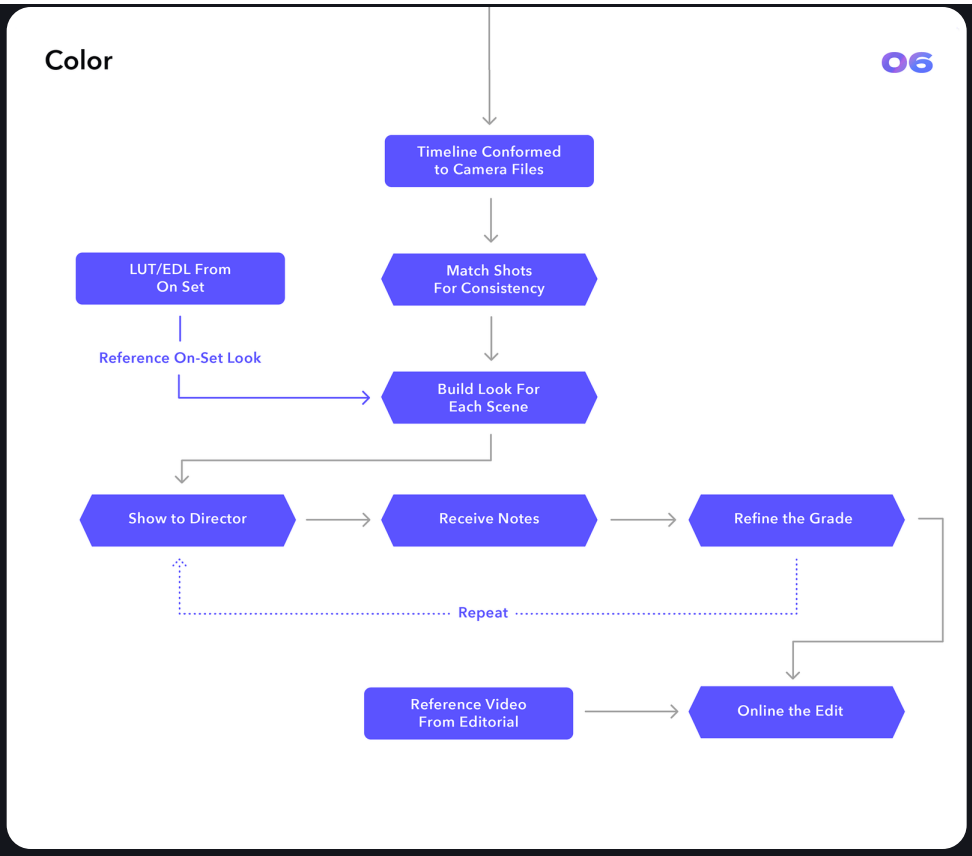


CHAPTER 06

# Color

The colorist puts the final polish on the film, using specialized tools to shift the audience's mood from moment to moment.

*8 minute read*




Once editorial and VFX have done their jobs, it's time to move on to one of the most important stages in the visual portion of the workflow: color correction.

Color correction is the process of adjusting and enhancing the visual characteristics of a film, including exposure, contrast, and color balance. Of course, various types of preliminary color work are done at earlier stages of the production, but color correction applies a finalized look that reflects the Director of Photography's artistic vision. This process is critical to both the technical and creative aspects of a film, as it ties together everything presented on screen into a cohesive and beautiful image.

## Digital Color

We've talked a lot about color so far. This is a very complex topic, and there are so many facets that it's easy to get confused. So, here is a systematic recap of digital color that

should help clarify what we're talking about.

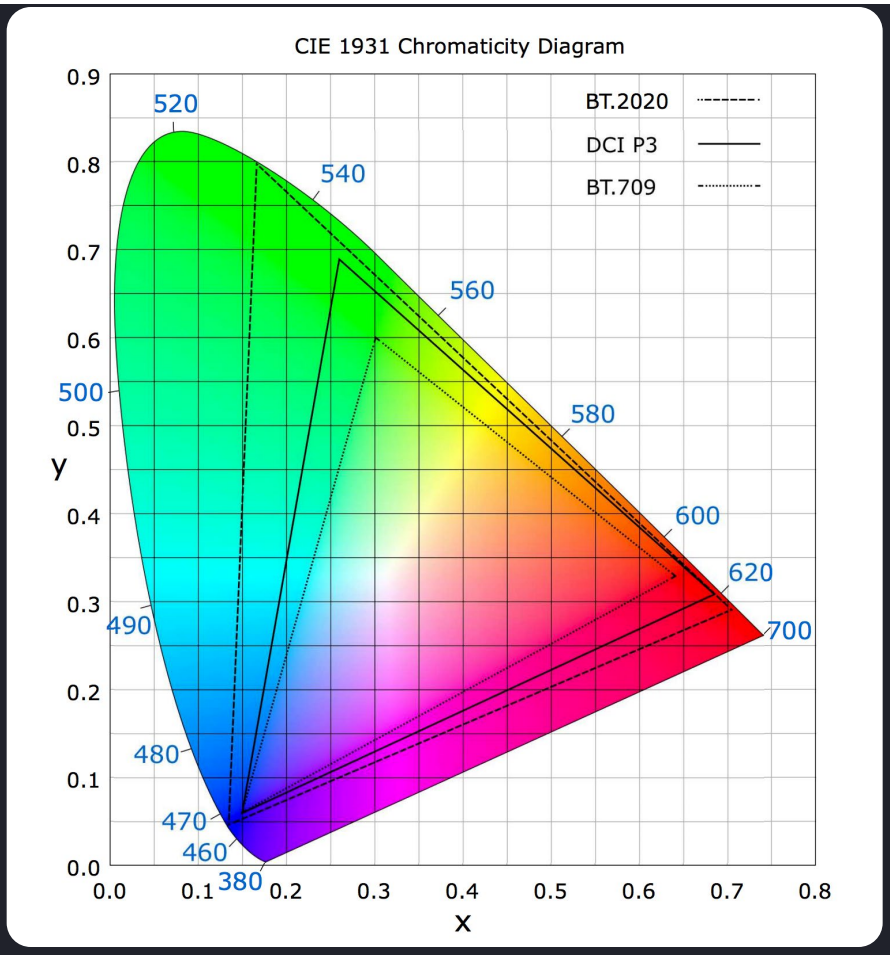
## Color

Color is a term that describes the human perception of different wavelengths of light. Our eyes contain one type of cell (rods) that is sensitive to light intensity (brightness), and three different types of cells (cones) that are sensitive to certain ranges of wavelengths (colors). Cones are sensitive to long (red), medium (green), and short (blue) wavelengths.

Of course, there is an infinite number of colors in the visible spectrum, and each of these wavelength ranges produces a multitude of colors beyond just red, green, and blue. Our brains represent these other colors by blending signals between wavelengths. These are how we "see" secondary hues like yellow, magenta, cyan, and every other color.

But our eyes can't actually differentiate the infinite number of possible colors. Based on real-world experiments, it's estimated that our brains can only differentiate around 10 million colors. This collection of colors is called the *spectrum locus*, and it represents all the colors an average person can see with their eyes.

These colors have been plotted onto a graph to form the CIE 1931 Chromaticity Diagram. This is the foundation from which digital color is built. Think of it as a map, where every perceivable color can be found.




But why are the colors plotted like this? In order to make this map, we needed a system for assigning locations to colors. And for that, we used a color model.

## Color Models

A *Color Model* is a mathematical method for describing a color based on its component properties. The standard color model for digital video is the RGB color model, which is the same model the CIE 1931 diagram uses. This color model describes every color based on the amount of red, green, and blue light that combines to create it. Why red, green, and blue? Because our biological vision system is based on those wavelengths.

But remember, we have a problem with digital color. Our

machines don't have eyeballs. To represent colors on digital machines, we need a way of counting them, because humans and machines can agree on mathematical values. So, if every color has a countable value, then we can tell our machines what to call different colors. That way, when we ask a machine for a particular color (by its countable value/name) the machine will give us what we want.

This is what the RGB color model enables. It plots colors based on their component parts—the amount of red, green, and blue that make it up. Mathematically, we can calculate these amounts as three percentage values. This group of three values is called an *RGP Triple*, and it's basically the location of a digital color on the map of the perceivable spectrum.

But if you tell a computer that a color is 25% red, 75% green, and 58.935% blue, it won't be able to make those calculations without more information. It needs to know what red, green, and blue points on the map it is supposed to calculate from, otherwise the percentage is meaningless. 25% of *which* red, or 75% of *which* green? Different primary color points will result in different outputs. To standardize the results, we need to tell machines which points to use, and we do that by assigning a color space to the image.

This is why, when you see color space diagrams like the one above, a color space is always a perfect triangle—it represents all of the values between three points.

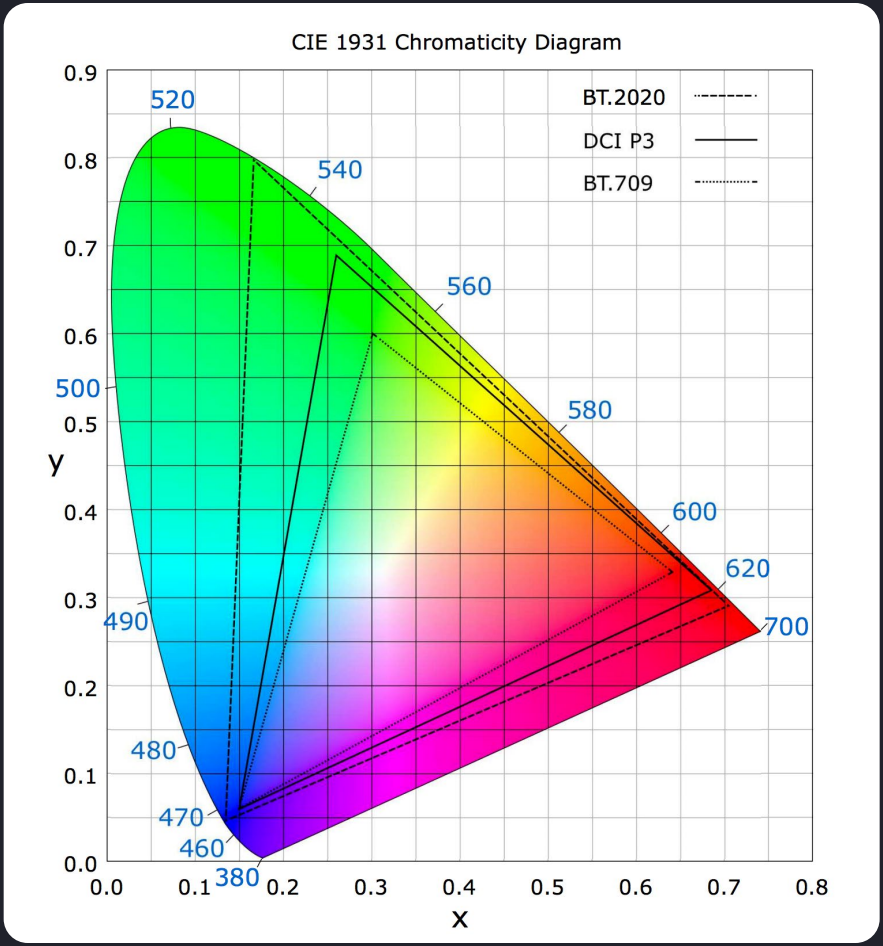
## Color Space

A *Color Space* is a specific organization of colors. It sets the location for the three primary colors, that is the points on the map that equal 100% of red, green, and blue. Once a computer knows the primary color points it's supposed to use,

it's quite easy to find every other color relative to them.

Now you might wonder why computers don't just utilize the points on the CIE 1931 color diagram. It is a color space, after all. Unfortunately, there are no machines that can actually display the total extent of the colors on the chart. It's just not possible yet with our current technology. So, in order to make it easier to use color digitally, we need primary color points that machines can actually handle.

The standard color space for most video is the Rec. 709 color space. It's a fairly small color space, but it has been the standard for SDR content for many years. As we move into the future of 4K HDR content, the larger Rec. 2020 color space will become the norm. There are also different color spaces for specific technical applications, like DCI-P3 for cinema projection.



The three triangles in this image represent those three color spaces. The curving shape that goes beyond them represents all of the colors that humans can perceive.

Each of the different color spaces has advantages and disadvantages, and comparing them is not always straightforward. One of the most important ways to compare them, however, is by comparing their color Gamuts.

## Color Gamut

A *Color Gamut* is just the footprint of a color space. It is the subset of possible colors that a machine has to choose from within a particular color space. If you take a color model and add a color gamut, you make a color space, which digital machines can understand. Generally speaking, a color space with a larger footprint is said to have a wider gamut.

It should be noted that a gamut is just the *extent* of colors that can be reproduced in a color model. It is not the *number* of colors that an image can contain. To make that measurement, we need to understand Color Depth.

## Color Depth

*Color Depth*, also known as bit depth, is a measurement of the number of possible colors in an image. It is determined by the amount of digital information used to describe every color. More bits of information allows for more possible colors to be described.

But color depth does not tell us which colors from what color space are in an image. To continue the earlier analogy, if a

color model is the map, and the color space dictates the addresses of every color, then an image's color depth determines how many addresses there are between each of the reference points (primary color points). So, it's sort of like the *resolution* of the map.

[Note: The terminology surrounding this stage of post-production can be confusing. The term "color correction" traditionally referred to a specific step in film development, but it's become common to use it to refer to the entire process of reworking the visual elements of video, as we do here.]

## The Tools of Color Correction

Like every part of post-production, color correction requires a number of specialized tools. In the film days, color correction involved vats of chemicals that developed footage with certain visual properties, but the complexities of digital video have greatly expanded the capabilities and responsibilities of colorists.

Digital color correction tools give you control over all of an image's properties, down to the pixel level. This precision opens up a vast array of creative possibilities. For example, you can alter the appearance of an actor's skin, replace an undesired color with an entirely new one, or even "relight" a scene after it has been captured. Knowing the basic components of these tools and what they can do is vital to understanding the artistic potential of color correction as a whole, and the impact it can have on your project.

The Tools of Color Correction

To plan artistically-sound color correction for your project, you first need to be familiar with the universal tools of the trade.

## Color

The colorist's most fundamental tool is, well, color. This might seem obvious, but digital color is complicated and needs some exploration.

When we talk about colors in everyday life, what we're really talking about are human perceptions. Our brain interprets a particular wavelength of light in a certain way, and we call that color by a particular name. Red, green, blue, periwinkle, etc.

But digital color is different. Since computers don't have eyeballs, the only way they understand color is based on what we tell them. Engineers and scientists had to develop complex mathematical schemes to instruct computers that one color is blue, another is red, and so one. How exactly this was achieved is a very complicated story, but it boils down to a few key concepts

## Digital Color

Digital color is defined in terms of fundamental properties that can be measured and calculated by a computer. There are different ways to talk about these fundamental properties,

but one of the most common ways is with *hue, saturation, and lightness*.

*Hue* is what people are normally referring to when they say "color." Red, green, and blue are the 3 primary hues that digital color is concerned with, plus the three secondary hues yellow, cyan, and magenta, which are created by mixing the primary hues together. All other colors can be made by combining different amounts of these hues.

S*aturation* is the intensity of a color. If a computer wants to display the color "red", it will also have to specify the question "how red?" A computer can answer this question with the saturation value of a color. The higher the saturation, the more intense the color. On the other hand, a color with no saturation will appear gray (have no color).

*Lightness* is a representation of how light or dark a color is. As you can probably guess, a digital color with a higher lightness value will appear closer to white, and one with a lower lightness value will appear closer to black.

These three fundamental properties can be used to describe any color to a computer. For digital video, they are separated into two parts of an video signal, called *Luma* and *Chroma*.

Basically, *Luma* contains the brightness information of an image (i.e. it's the black and white part of the video signal). So, that means it holds the lightness values of every pixel.

*Chroma* contains all the color information of an image, including the hue and saturation for the three primary color channels of every pixel.

This is a radical simplification of digital color, but it illustrates the complexities you deal with when you talk about color in your project. It's a foundation from which you expand and

finesse your creative and technical skills.

## Manipulating Color

Now that we have a basic grasp of how digital color works, we are better prepared to use it for our artistic purposes. But how exactly do you manipulate color for your project?

Perhaps the most common tool for working with color is the *Color Wheel*. This isn't exactly the color wheel you remember from grade school art class, though they do look similar. Color wheels let you quickly adjust the hue, saturation, and lightness of colors at the three *tonal ranges* (that is the blacks, midtones, and whites) of your image. This makes them a powerful tool for a wide array of color correction tasks.

Another option for manipulating color is the *Curves* tool. This tool presents a simplified version of a video signal as a diagonal line in a graph. By "curving" the line within the graph, you can rapidly adjust the color balance, brightness, and contrast of an image. While curves are a lot less intuitive at first, they can still be used for very precise color correction in your workflow. In some cases curves will be your first choice, either for the sake of speed or simplicity.

Deep Dive: Color Wheels and Curves

There are a great many more tools for working with the color in your project, but color wheels and curves are arguably the most universal. Pretty much every piece of software capable of image manipulation has some form of these tools.

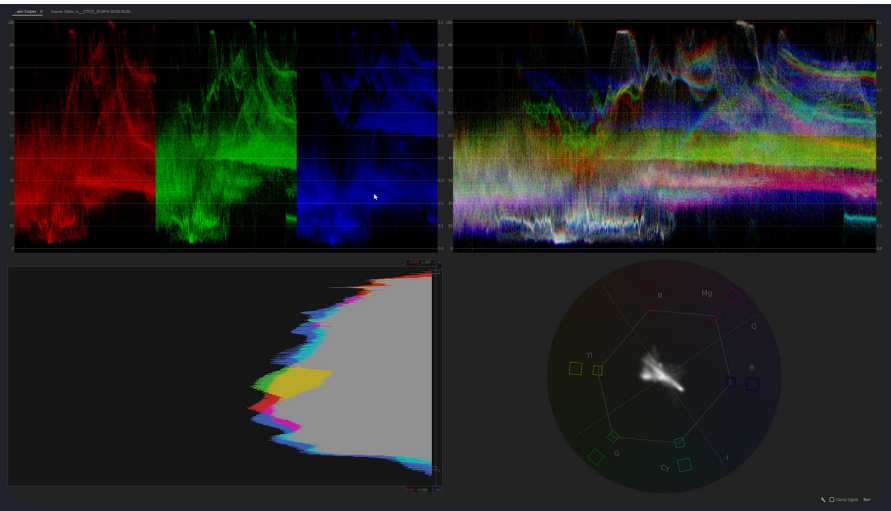
# Evaluating Color

Manipulating color is tricky business. Video monitors (even the same model) often vary in how they display colors. In theory, properly calibrated displays fix this issue, but in reality they can still drift apart over time.

Even if you re-calibrated your monitors every day, there's another problem. Human eyes aren't consistent either. Your eyes (and brain) adapt your perception of color depending on a number of factors, including the lighting of the viewing environment, the amount of motion in an image, and even your level of stress. Put simply, color is hard because your equipment lies, your eyes lie, and your colleagues lie (because their eyes lie too).

But scopes never lie. Scopes are specialized tools that measure exactly what is contained in a video signal. They measure this information, and then visualize it onto a type of graph. There are several different types of scopes, and each uses a different graphing method to visualize the video signal for different purposes.

In general, most scopes help you understand some aspect of the color and/or brightness information in an image. So, even if your monitor *displays* wildly inaccurate color, scopes *measure the source information* of a video signal. That means you can always get a clear representation of the actual visual characteristics in your footage.



The best part about scopes is that they are universal. No matter what hardware or software you use, they always show the exact same values for the same video signal. They are a fundamental tool, so you really need to know how to use them when it comes time to color correct your project. Without them, it will be difficult to know if you're crushing the blacks in an interior scene, clipping the whites in clouds, or if you have the correct hue and saturation for your customer's logo.

They really are the only way to accurately evaluate the colors in your video signal.

Deep Dive: Video Scopes

Scopes break down all the pixels of an image into luma and/or chroma information and then visualize it so it can be easily read. This visualization is called a *Trace*. Depending on the function of a scope, the trace will represent different aspects of an image and be displayed in various ways.

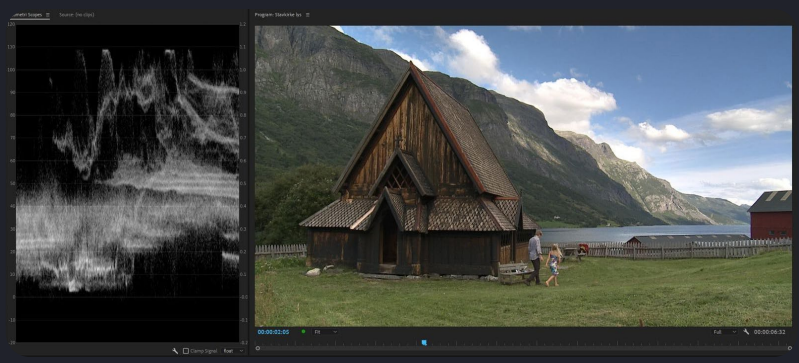
There are three general ways to trace a video signal on a scope–with a *Waveform*, *Histogram*, or *Vector Scope*. Within each of these categories there are numerous

configurations, with over a dozen different scopes to choose from (depending on how you count). Thankfully, there are only a few primary video scopes that you are likely to need for your project.

## Waveform Scopes

Waveforms are often the first scope you'll use in the color correction process. They graph the visual information of an image onto x and y axes, with the vertical axis measuring *values* (of the scope), and the horizontal axis representing *position* (in the image).

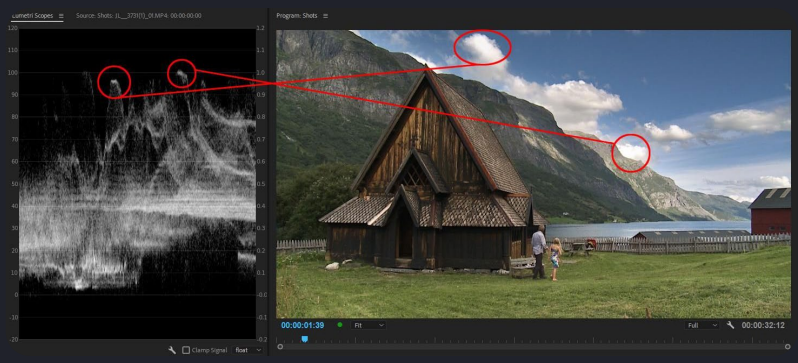
The most common waveform is the *Luma Waveform*. As the name implies, it measures the brightness of an image.



Luma waveforms measure the pixels of an image from dark to light (bottom to top). Bright pixels of the image are plotted higher in the waveform, and dark pixels are plotted lower. These brightness values are then arranged based on their actual horizontal position in the frame (left to right). So the furthest left part of the waveform is the furthest left part of the image. The same is true for the furthest right part, the middle, and every other horizontal position in the image. In short, left is left, and right is right.

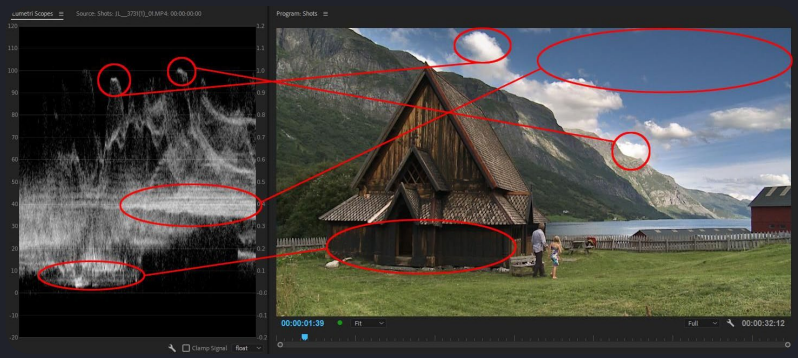
But up is not up, and down is not down. Remember, the vertical axis only measures brightness. So then how do we read it, exactly?

When the waveform traces the image, it's actually looking at 1-pixel-wide vertical slices. The pixels in every vertical slice are arranged vertically based on brightness. So a bright pixel at the bottom of an image will appear at the top of the waveform. To read this scope, look for the brightest peak in the image, and then try to find which part of the image it corresponds to (usually the purest white on screen).




Note how the position of the spikes in the waveform mirror the position of the brightest pixels.

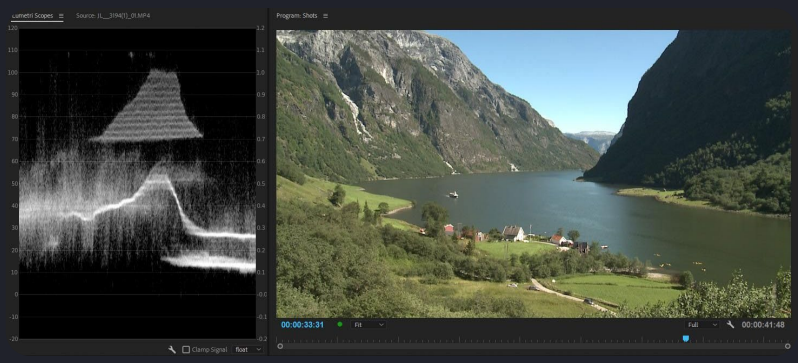
Here we've circled the corresponding areas in the scope and the shot. Now you can clearly see that the brightest peaks on the waveform are located at the same horizontal position as the clouds in the shot (roughly the middle).



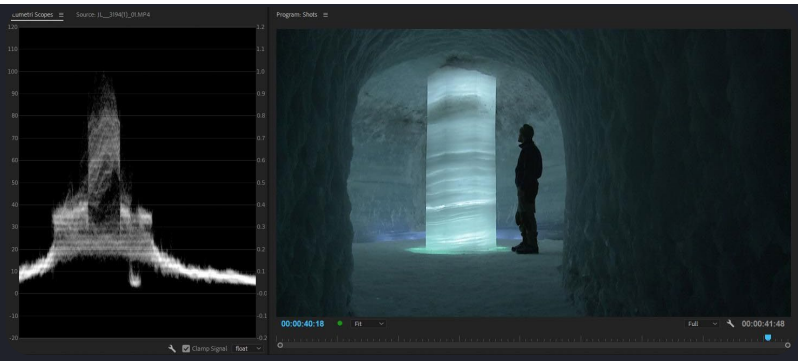
The area with lots of dark pixels on the left is the dark wall of the church. The area with lots of midtone pixels to the right is the blue sky (and probably parts of the green grass below).

And here we can see the darkest parts of the image (the front of the church) are at the same horizontal position as the lowest part of the waveform trace (in the left/middle of the frame). The big patch of open sky in the upper right of the frame is neither super bright or super dark, and so it's fits squarely in the midtones, which mirrors the waveform trace.

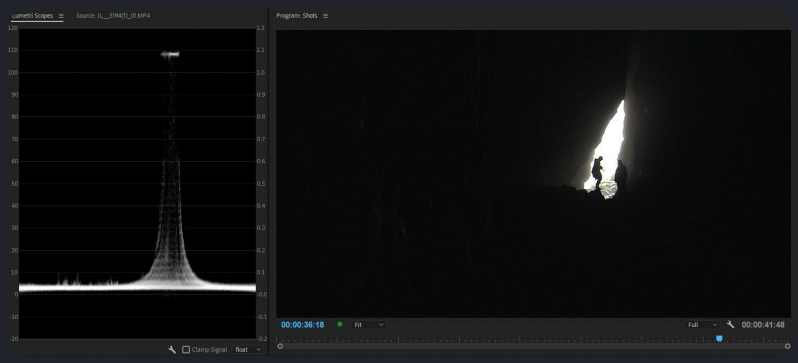
It can take some practice to read waveforms quickly, but you start to manipulate an image and see the scope update in real time, it will make a lot more sense. Here are some other images that show how the distribution of light and dark pixels from left to right is mirrored in the waveform scope.



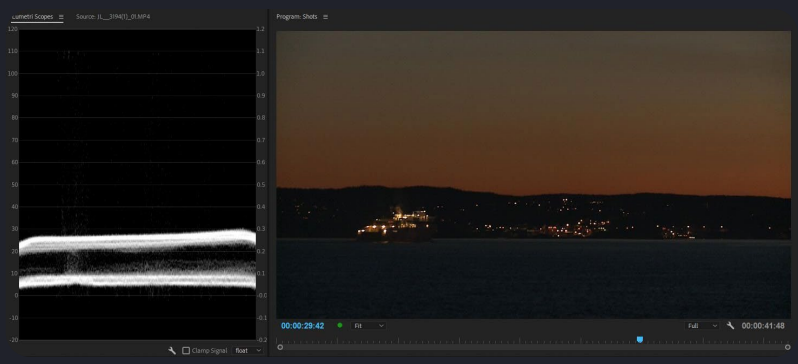

Scope shows lots of dark pixels on the right, some bright pixels in the middle, and mainly midtones on the left side.




Scope shows dark pixels on both sides, some midtones in the middle, and a narrow area of bright pixels a bit to the left of the middle. Can you spot the man's dark silhouette on the scope?



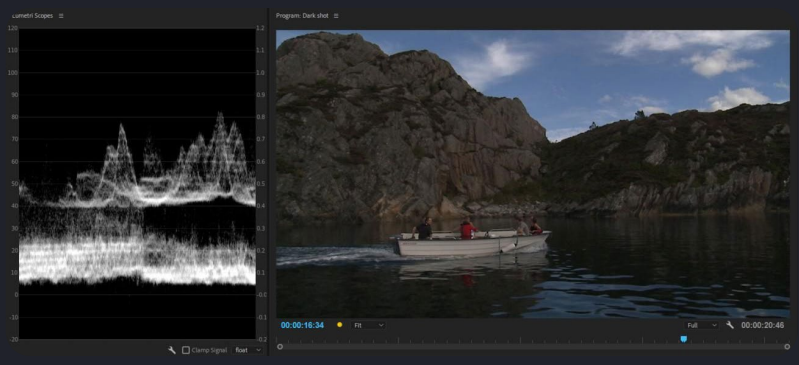

Scope shows lots of black or near black pixels across the whole image, and a small area of very bright pixels a bit to the right of the middle.




Scope shows very dark pixels across the whole image, and some brighter, but still pretty dark, pixels, also across the whole image.

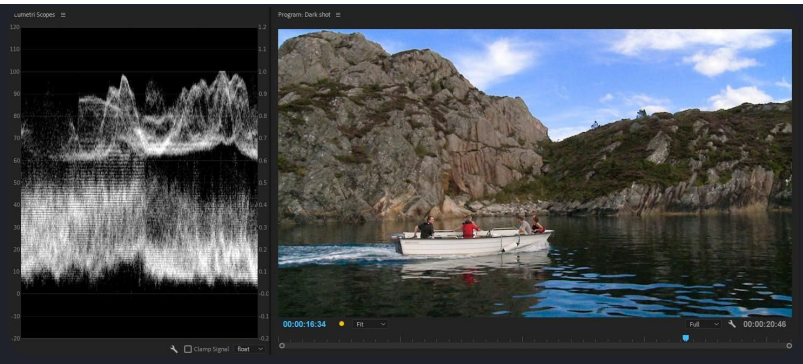
As you can probably guess, the Luma Waveform is an excellent tool for checking the exposure of your footage. The numbers on the scale are measurements of brightness for particular video standards. In this case, it's for HD video signals, where absolute black is 0 and white is 100. Any part of the signal that falls below 0 will be clipped, as will any level above 100. If you're color correcting your project for broadcast, keep your levels within the 0-100 range.

[Note: This is true only for Standard Dynamic Range (SDR) video. Newer Ultra-HD standards have introduced High Dynamic Range (HDR) capabilities, where whites can go all the way to 10,000. This will be covered in another section.]




Waveform Scope for an underexposed shot

In this underexposed shot, we can see that the the brightest area of the image peaks at around 80 on the scope. Remember, "brightest" means "closest to absolute white." For this shot, that is the clouds. However, since the signal at those points only reads 80, it means the clouds will not be fully white. The darkest part is around 5 on the scope, which means no part of it is fully black either. So, the scope tells us that nothing is being clipped. This is really good information that will help correct the exposure.



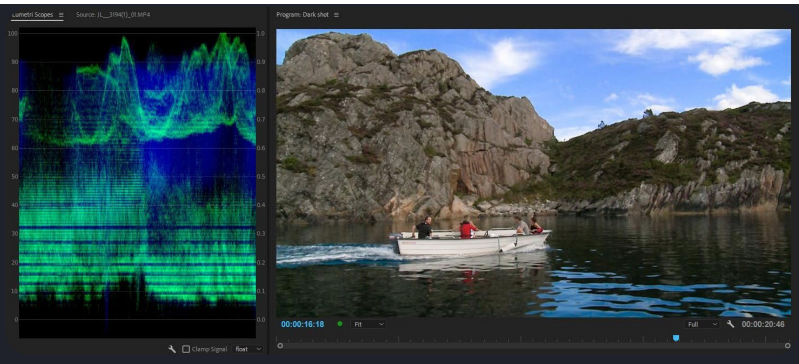

Waveform Scope after the shot is corrected

By raising the white levels, slightly lowering the blacks, and making the midtones a lot brighter, the image looks a lot better. The biggest difference before and after correction is that the midtones and shadows are spread over a larger range in the waveform. This gives the image much more contrast, and brings out details we couldn't see before.

Of course, not every image will have exposure that stretches perfectly from 0 to 100. That said, using a Luma Waveform to correct exposure is usually the first step in the coloring process.

But it's called *color* correction for a reason. We need scopes that tell us about the chroma in our images.

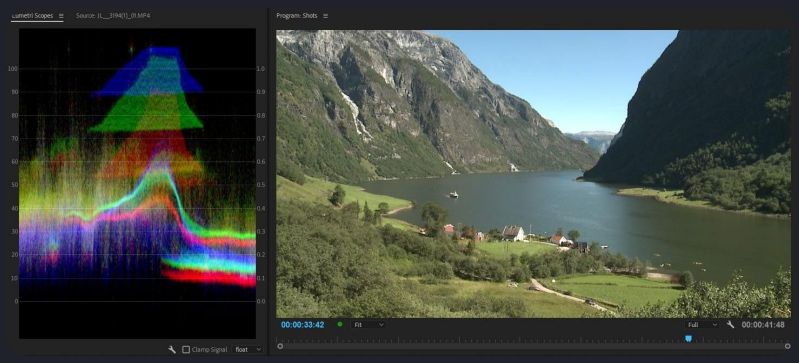
Well, the simplest scope that displays color information is the *YC Waveform*. This scopes displays both luma and chroma information of a video signal, where the two parts are overlaid and shown in different colors.



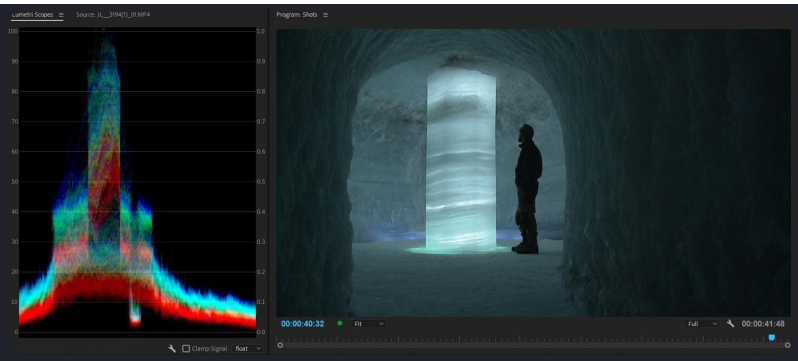

The YC Waveform Scope shows the chroma signal (blue) as an overlay on the luminance signal (green).

However, this is not the most useful scope, and you'll probably use the chroma overlay much less often than other, more useful scopes.

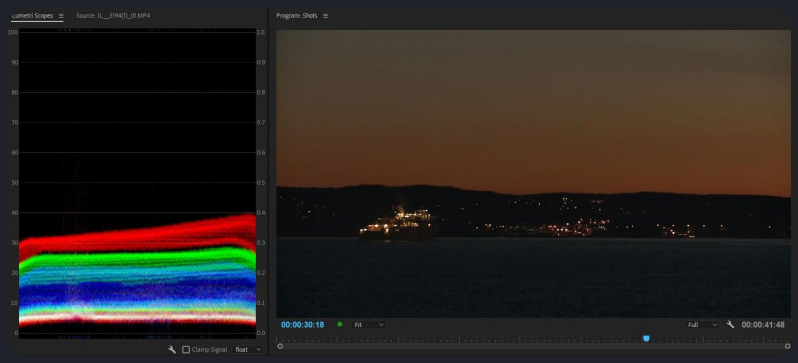
The better solution for viewing chroma information on a scope is the *RGB Waveform*. This scope is almost identical to the luma waveform, but it breaks down an image into the three primary chroma channels, not just the luma.




Here, we can clearly see that the dark pixels on the right have a lot higher levels in green than in red and blue. And the area with bright pixels in the middle clearly has mostly blue, but also some green – and very little red, compared.

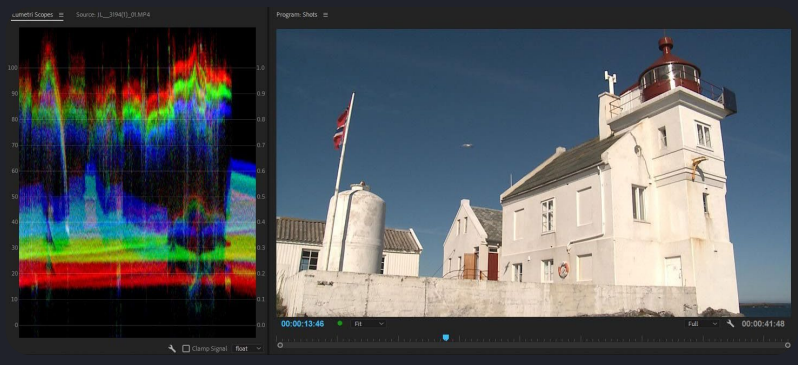



Now we see that this image has more blue and green than red. The blue and green pixels are at about the same level, creating a cyan area above the red area in the scope.




The RGB Waveform clearly shows that the darkest areas are pretty neutral, creating a white stripe at the bottom of the waveform. Plus, it also shows that the brightest pixels have a lot more red than green, and even less blue – so it's an orange color.

The RGB waveform is very helpful for identifying color casts in an image, and neutralizing them.



Red, green and blue bands show that there are unequal amounts of the three colors.

In this image, the lighthouse is a bit warm, as you can clearly see in the scope. There are distinct bands of red, green and blue in the brightest part of the image (the white walls).

When the bands overlap and show as white, the walls are neutral.

This is easily corrected by adjusting the adjusting the color channels in the top tonal range so that the distinct bands overlap. Now that they have been corrected, they form a white streak in the top of the waveform.

## RGB Parade

As useful as the RGB Waveform is, it can be confusing to differentiate the three color channels in a single scope. This is where the RGB parade comes in handy. This scope arranges the waveform for the three color channels side by side, rather than stacking them on top of each other. With it, you'll be able to much more precisely balance your image's color.

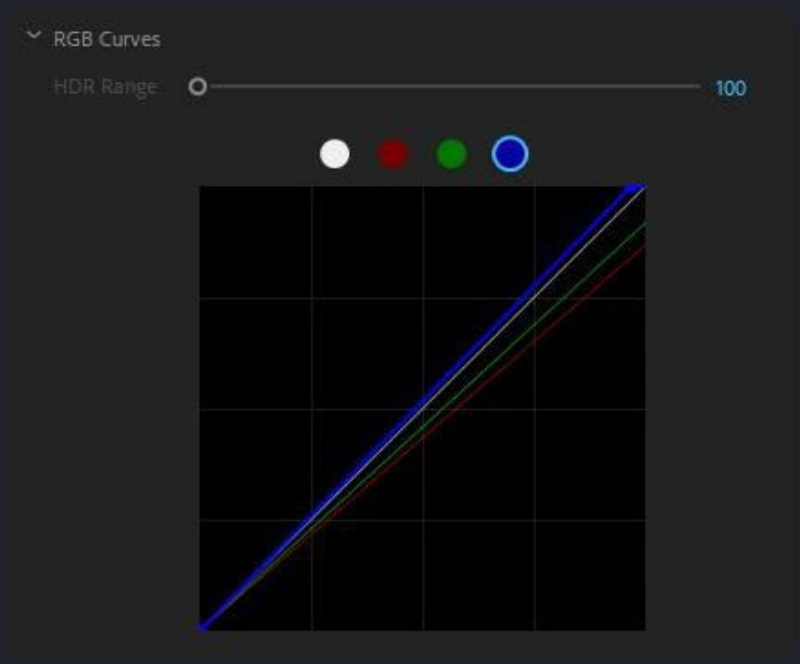
The same light house image, and in the RGB Parade Scope we can still clearly see that the highlights have more red and green than blue, so they're warm. We can also see that the midtones have a lot more blue than green and red, making the sky blue-cyan.

After correction, we see that the highlights are pretty much equal, making them neutral, and the midtones have more blue and a bit less red, making the sky a deeper blue.

The RGB Parade scope is easily one of the most useful scopes, as it allows you to quickly spot color balance issues. This is corresponds very well to the RGB Curves, which is able to quickly correct color balance issues. They're a very good pair to use together.

In the above image of the lighthouse, the color imbalance can easily be fixed by dragging the upper point in the curve for each of the three primary color channels. This is a very quick method. Adjusting the

blue curve only affects the blue waveform, the red curve only affects the red waveform, and so on. It is an intuitive way to balance color. Too much red in the highlights? Drag the top of the red curve down. Not enough blue in the shadows? Drag the bottom of the blue curve up. It's really that simple.

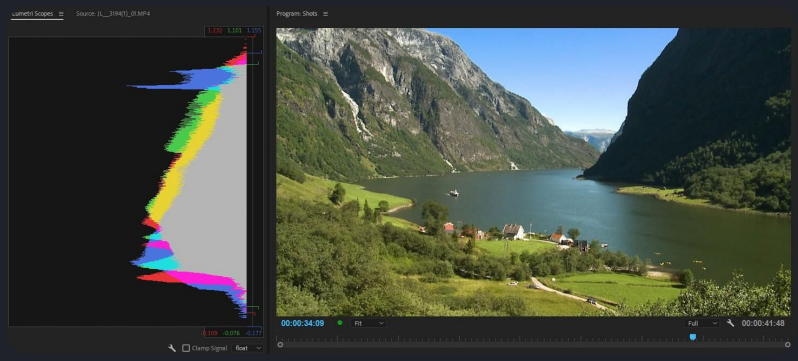



The RGB Curves adjustment needed to raise blues, and lower greens and reds. This screenshot is from the Lumetri Color Panel in Premiere Pro.

As you can see, the various flavors of waveform are very useful for evaluating your image and determining what needs to be corrected. They give you a quick, clear representation of what your image looks like, and which parts of the image need work. The Luma Waveform and the RGB Parade will be some of the most valuable tools available for color correcting your project.

## Histogram

The *Histogram* will probably look familiar to anyone who has worked with almost any sort of camera or image editing software. A histogram displays the distribution of dark and bright pixels in an image, but (unlike the waveform) does not tell you where those pixels are in the image. This makes it a bit less useful than the Waveform Scopes when color correcting video, but can be useful in certain scenarios.



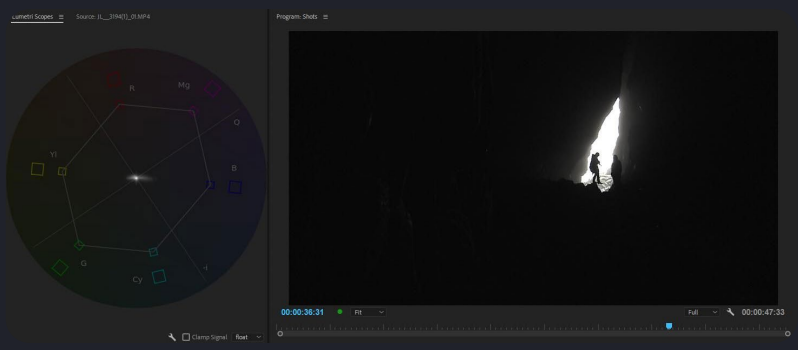

With separate red, green and blue histograms, you get more info on the levels of each color.

For example, this histogram (in Premiere Pro) shows the minimum black and maximum white levels as numbers for each color in the image. This allows you to quickly identify peaks that may escape your eyes in the Waveform Scopes (because the peaks are too small to see).

This makes histogram a useful tool for evaluating exposure, color balance, and illegal luma levels. Just like waveforms, histograms can display the luma signal by itself, or separate the chroma signal into individual scopes for red, green and blue. Some histogram run vertically, some run horizontally, but they work the same way. The trace measures the volume of pixels at each luma/chroma level from darkest to lightest.

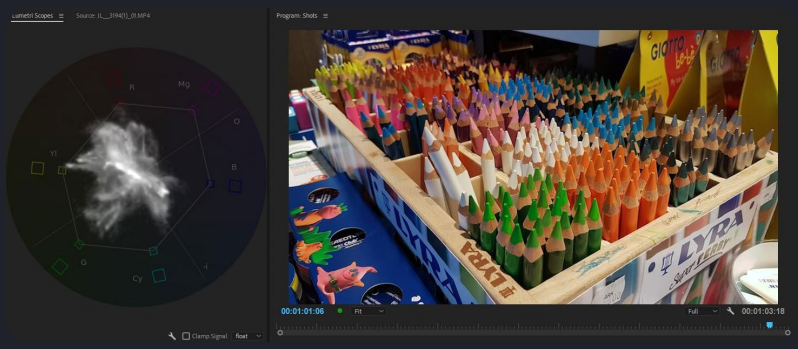
## Vector Scopes

The last scope you will use most frequently is the *Vector Scope*, specifically the YC Vector Scope. Vector Scopes measure the hue and saturation of colors in an image. The trace is drawn within in a circle, with the distance from the center indicating the saturation of a given color. The further from the center, the more saturated the color. The center of the scope indicates white (no chroma data), so if you only see a single dot in the middle of the scope, that's a black and white image.




An image with very little color creates a small dot in the middle of the Vector Scope.

The angle of the trace indicates what hue the pixels in the image are. The brighter the trace is at a particular angle, the more pixels in the image are that hue.

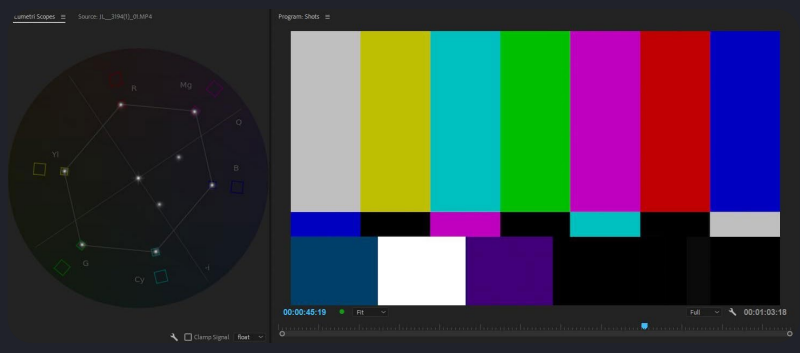



The saturated colors of the colored pencils make the trace go wide, and because there are so many different

colors, it's spreading in many directions.

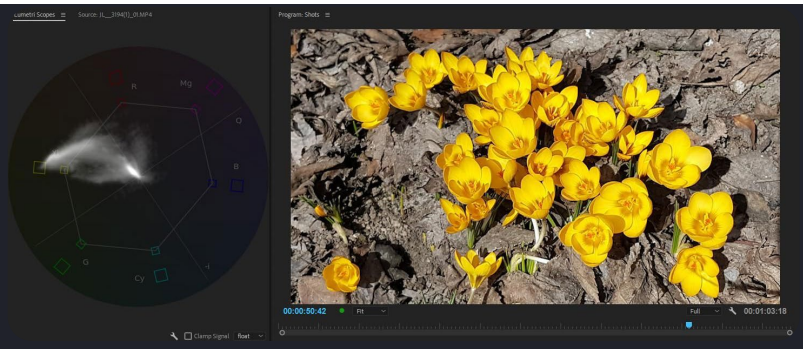
As we can see in the above image of the colored pencils, the variety of vivid hues on screen makes the trace stretch toward several different color targets. The trace does not correspond to the location of these pixels in the image, just their chroma information.

This makes vector scopes the best way to spot oversaturated (illegal) color in an image. You don't have read any other information in the trace, just what colors are present.




The trace from a colorbar will show dots in the inner boxes. The two dots that are not in a box are the two colors in the lower row.

The classic color bars produce a signal with the maximum-allowed saturation for each of the hues. The trace it produces clearly illustrates the limits for broadcast safe color, as shown in the color targets (represented by the boxes surrounding the center). Each one is labeled for their hue—B for Blue, R for Red, Yl for Yellow, Cy for Cyan, etc.




There's a lot of yellow in this image, so the Vector Scope lights up in that direction. Since the trace goes outside the lines between the boxes, this image will be rejected by some broadcasters. The yellows need to be desaturated.

Any chroma signal that goes beyond these boxes (or the lines that connect them) is considered illegal, and so should be desaturated to meet the standards.

But vector scopes are also very useful in other ways. They are a great tool for making sure your skin tones are accurate. There is a line, extending from the center to the midpoint between the red and yellow targets, where the "ideal" hues for skin tone are found. If you want the onscreen talent to look healthy, make sure the hues of their skin fall as close to the line as possible.

Footage courtesy of Sean Fulton and Martin Fotland. Model: Marianne Melau.

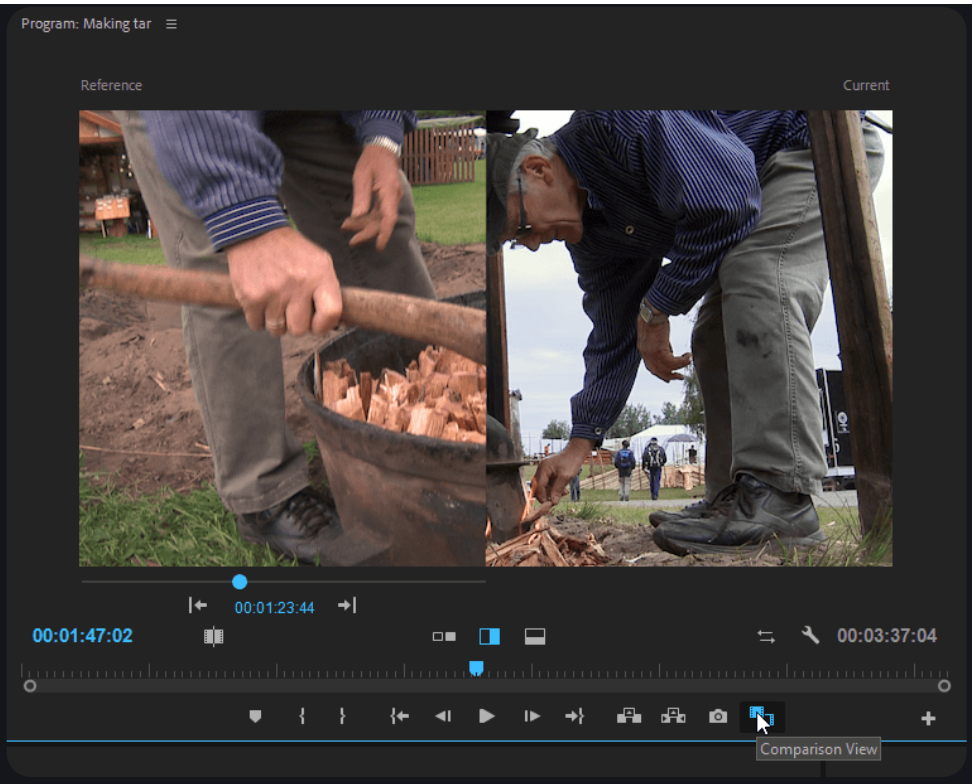
So a vector scope can be a much more clinical tool than waveforms and histograms. It allows you to rapidly check your image for certain color values, and then make adjustments based on predetermined technical standards.

### Other Scopes

Once you've mastered using these scopes, you're bound to discover more hiding within the menus and panels of your NLE or color correction application. Many of these scopes will likely be specialized tools for video engineers or network quality control technicians. The average editor or colorist will not find them as useful for creative work. However, now that you know how to read the most common video scopes, you should be able to handle the many derivations that exist.

## Creating Visual Continuity

The first step of the color correction process is to make sure that all of the shots in a single sequence work together visually. This is often also called shot matching. In an ideal world, a DP would work out everything on set so that all the footage from a scene matched perfectly. Unfortunately, reality is rarely so kind. Even if you use the same cameras, lenses, and lighting for every scene, it is usually impossible to have a completely uniform color in all your footage.




A colorist will start by balancing the visual attributes of a film, shot by shot, to create a seamless baseline of color. This eliminates many of the visual differences between takes, locations, and cameras, and ultimately serves to maintain a story's continuity.

Nailing down this baseline is the first priority of color work. It might be less exciting, but it will save a lot of time and frustration later on.

## Shot Matching and Primary Color Correction

The main process colorists use to achieve visual continuity is called *Primary Color Correction*.

Primary color correction involves manipulating the entirety of an image. If you change luma or chroma values with a primary color correction tool, every pixel will receive those adjustments. This is why primary correction tools are so useful for shot matching, because they can apply similar changes to different images in roughly the same way.

Typically, the first step of primary correction is to set the luma (brightness) to a desired level. You do this by adjusting the three tonal ranges until you achieve the appropriate contrast or exposure.

The second step is to adjust an image's chroma (color). This is your opportunity to fix color balance issues, like an incorrect white balance or an add color cast.

After the color channels are balance as intended, you can adjust the saturation of the various hues according to the baseline look you plan to give the shot.

These are the three general steps of shot matching. Though shot matching is not always achieved using just primary color correction processes, generally speaking they are all you need to establish a baseline of color.

That makes primary color correction a particularly technical process. Colorists are not so much making artistic decisions as they are measuring and adjusting specific aspects of an image, at this stage.

Scopes are vital to every phase of color work, but they are particularly useful here. From the start of you color work, you will know what the intended destination of the project is, and can thus make make primary adjustments based on the relevant standard. For example, if your project is headed for broadcast, then you know to set your luma and chroma levels within the legal limits of Rec. 709.

## The Process

Shot matching is not just a piecemeal process of adjusting every shot in your timeline sequentially. Rather, there is a

recommended method that will save you time and effort.

The first thing to do is to pick a *Hero Shot*. This is a shot that best represents a scene's color and exposure. Often times this is a wide shot that encompasses many of the details captured in the rest of the scene, such as faces or environments.

Not only is it important that the hero shot be representative of the rest of the scene, but it also must be technically sound. If a shot is massively over exposed or was recorded with a strong color cast (compared to the other shots), then it's not a good candidate for building a baseline of color. That doesn't mean it needs to be the best shot, but it can't be a bad one. In general, choose a hero shot that is as similar to as many other shots in your scene as possible.

This is where your primary color correction begins. As laid out above, you should adjust your luma levels for the tonal ranges, make sure the color balance is neutral, and set the saturation so that skin tones and recognizable colors (like foliage) look normal.

Once you have done this for the hero shot, you can start matching the other shots in the scene to the hero using a comparison window (a special view within your software that will put two images and their scopes next to one another). After all the shots in a scene have been matched to the hero, then the process begins again with the next scene.

This is how you build a baseline of color across your whole sequence. One very useful benefit of establishing the shared baseline is that you can adjust all the shots together, rather than having to do them one at a time. For example, if you match 30 shots to a hero, but later decide you want a different look as your baseline, you can just apply the same global changes to all the shots using an adjustment layer, node, nested sequence, or whichever mechanism your NLE/color

software utilizes.

Primary color correction tools are vital to the color matching process, but are also used to build creative looks after the shots have been matched and grouped.

## Crafting The Look

Once the baseline of color correction is complete, the second task of a colorist is to craft the artistic look of the film (sometimes people use the term "Color Grading" to refer specifically to this stage). This is the process of using color to convey a certain atmosphere or emotion in your story. How exactly the film is colored will vary depending on the genre, subject matter, and style, but it will fundamentally influence the viewing experience.

### Secondary Color Correction

A very important part of creating the artistic look of your film is drawing the viewer's eyes to particular aspects of an image (faces, props, logos, and lots of other things). Now, of course, much of this work is achieved by the DP during shooting. Shots are framed in camera so that the most important elements of a scene are prominent, while less important elements fall into place around it. But sometimes you'll need to alter an image even more precisely than pure cinematographic techniques can achieve. This is where secondary color correction comes into play.

Secondary color correction is the process of manipulating specific parts of an image. Unlike primary color correction

where every pixel on screen is affected by the changes, secondary correction only adjusts certain area within a shot.

There is a multitude of tutorials online that use secondary color correction to radically change the color of someone's clothing, or perhaps completely redo the color of a set/environment. However, in most cases, secondary color correction is used to much subtler effect.

One of the biggest uses of secondary color correction is to adjust skin tones, and other colors that humans have a common familiarity with (like natural colors in foliage and the sky). These colors are often less than perfect after a round of primary color correction. After all, primary tools are most concerned with getting the majority of the image where it needs to be. But that can be to the detriment of specific parts of an image. So, secondary tools help you finesse those colors to look just right.

But other than simply "fixing" what primary correction leaves out, secondary correction enables you to hone a shot for specific purposes of the story. Want to build tension over a particular part of the screen? Darken everything around that area, so that it ominously stands out. Need a character to feel a bit happier? Add a subtle glow around their face. The potential to use secondary tools to craft the look and feel of a scene are almost limitless.

## Making Secondary Corrections

To make a secondary correction, you must first isolate the part of the image you want to adjust. There are three ways you can do that.

The first method is to select pixels based on their value. This

value is usually a specific Hue, Saturation or Lightness (or combination of all three), a process known as HSL secondaries. To use this tool, you set a range of *qualifiers*. Qualifiers are usually in the form of slider bars, where you set two points along the line, and the values within those points are the selection. In essence, this selection is a mask defined by HSL color values.

HSL Secondaries are essentially the same as Keyers, which we will cover in more detail in the VFX section.

The second method to isolate part of your image is to select pixels based on their position in the frame. The tool for this is often called a "power window" but you can just think of it as a spatial mask. These selections are drawn on the screen by the colorist around the pixels they want to manipulate. Most often, these are just simple geometric shapes like circles, ovals, or rectangles, but they can also be complex shapes that completely encompass particular set pieces or characters.

The third method is to select pixels based on a combination of their values and position. This is very useful when neither of the above methods is quite precise enough in selecting the exact pixels that need to be adjusted. Heavy color work will chose this method most often.

For instance, it's common to want to select and change just a character's skin tones, and an HSL secondary is a great way of doing that. In some cases, however, there may be other objects (perhaps a cream-colored wall) that are almost the same color. An HSL secondary would affect both the skin and the wall. Adding a power window around the character could prevent the correction from touching the wall as well.

Keep in mind that there are tradeoffs to each of these methods. Selecting pixels based on value qualifiers is fast, and doesn't require any motion tracking to maintain the mask, but

is often less precise. Selecting pixels based on their position is more precise, but often requires motion tracking, which can be slow. The third method of combining value and position is the most precise, but takes the longest. You will likely use all three methods when coloring your film. But it will just depend on the shot.

One you have your selection, you can adjust all of the properties of the pixels within it using the tools we learned before, like color wheels and curves. Pretty much anything that you can do to color in primary correction, you can do to just the isolated parts of your image in secondary correction.

Good color correction pushes a story's tone and captures a certain emotional reaction from viewers. Even subtle changes can have a big impact. For example, changing the lighting on a character's face can alter how viewers perceive them. Likewise, recoloring the sky in a scene can shift the tone of an environment to lighten the mood, or to make it heavier. However you color your film, taking the time to be thoughtful and deliberate can make your story all the more captivating to audiences.

Deep Dive: The Psychology of Color/Color by Genre Style

While it's necessary to talk and think about color as physical entity (a certain amount of light falling on a certain kind of surface creating a certain value that can be mapped to a certain color space), it's important to remember that human beings are not perfect observers of color. We've all had the experience of two people describing the same color differently ("the blue car, over there," one person says, while the other corrects "you mean the green car?"), and that's typical even among people who have accurate "full color" vision. Once you add in variations in human vision (for example color blindness, which has a great variety of its own),

the tricks of memory, and the power of associations the psychology of color can be a tricky rode for a filmmaker to navigate.

We want to discuss here not just the subjective human nature of color perception but also the cultural and narrative associations we create with color that make the process of effectively using color in your film even more powerful once you understand the goals.

## Human Color Perception

The first thing to understand is that the human visual system is more complicated than a simple manual camera. It doesn't neutrally reproduce what it perceives, but processes visual information before you are aware of it. Much like a modern digital camera, it is constantly working to adapt itself to whatever environment it is in in order to see as clearly as possible, which can often lead to mistaken perception when you don't fully understand how the system is operating. On the flip side, once you have an understanding of how humans perceive and analyze color, it's possible to deliberately manipulate it for the benefit of your project and can be a powerful tool in your arsenal.

The brain is constantly analyzing the visual information it is getting and attempting to create a neutral image. This is because in the natural world human beings encounter a wide variety of lighting situations that create a color cast that can interfere with properly distinguishing elements in a scene. Daylight in an open field is very bright light, and very blue. Dusk in the woods is much darker, and oranger. Oranger still is the light of a fire at night time. However, our brain renders all of those to "neutral" colors in order to give ourselves the best chance to analyzing the information in front of us.

## Complementary Contrast

This phenomenon is often first appreciated when introduced as an optical illusion in childhood. The classic illusion is one where you stare at a red square on a piece of paper for more than a minute, then look at a wall and magically a cyan box appears is a result of your brains image processing. Your brain sees the blank red box and adds cyan to your vision to attempt to "neutralize" the image, in the hopes of balancing it out to reveal more information through contrast.

Your brain is constantly doing this while watching film and video entertainment; it's part of the process for how your brain neutralizes color casts out of various lighting situations. Thus, if you make a scene or sequence exceptionally red, your brain will gradually add more and more cyan to your vision in order to "neutralize" the red away and allow you to see the image from a more balanced perspective. There is a technique to control for this, and that is the use of a contrasting color. Color has an interesting phenomenon of complementary contrast. This means that human beings have a tendency to see colors as being more intense, or more saturated, when placed against a complementary color.

If you have a blue can of soda, as an example, and you place it on a blue table, it will look less saturated to your eyes than if you place it on a red table. The simultaneous contrast between the colors helps push both colors "away" from each other and make for a more intense image.

If you have a red field with cyan dots in it, your brain won't add cyan since it is already there: a wide variety of visual information is present and your brain doesn't need to try and neutralize out the color cast. The presence of complementary colors has the power of preventing your brain from neutralizing a color and also creates the impression that a

color is more intense than it appears without its complement nearby. There are three primary complementary pairs (red and cyan, green and magenta, and blue and yellow), and all have this effect.

In general it is a good idea, if you want to maintain the feeling of a color, to have elements of its opposite. While a film like "Eternal Sunshine" is quite well know for the blue cast to it's images (which fit the melancholy mood created by the story, production design and music), if you watch the film closely there is often a yellow reference in frame, such as a vase, a painted bollard, or another prop that helps create the contrast that keeps the blue images blue. Without those yellow elements your eyes would quickly "tune out" the blue creating more neutral images, which was not the intent of the filmmaker.

Of course, there might be situations where as a filmmaker you want to tune out a specific color. For instance, if you want to end your film with the theft of a famous emerald that will be greener than any audience has ever seen, you could shift the scenes leading up to the emerald's reveal more and more magenta. This will cause the audience to "add green" to their vision, so that when the green emerald appears (sitting on a magenta velvet pillow, of course), it will be far greener than if it hadn't been preceded by several scenes preparing the audience for that intensity. By the end of the "magenta" sequence the images might not appear magenta at all to an audience who has gotten used to it, but it well set them up well for a rich green in the next sequence.

## The "Meaning" Of a Color

Of course in many productions there is an emotional meaning that is attached to color. Take the classic Steven Spielberg film "Raiders of the Lost Ark." Red in that film means "danger." It is

first introduced when Indiana Jones reveals his pistol against a red carpet while packing for travel. It continues with the red pants Marian wears when she's kidnapped. It's of course echoed in the red Nazi flags towards the end of the film as Indie prepares to rescue Marian and attain the ark. It is a clear, distinct association in the film, undercut only by the red lines used on the map to indicate flight paths, which doesn't forebode danger, but doesn't distract too heavily from the association. Whatever reason Spielberg had for those red lines (perhaps tradition), they don't undercut the association since Spielberg quickly shows us a Nazi reading "Life" magazine with it's signature red logo, to keep up the connection.

You will find many discussions from various artists about the inherent meanings of a given color. In truth, while many would argue that these associations are "core elements" of a color, that simply doesn't seem to be the case when observing real world filmmaking. Yes, "red=danger" in RAIDERS certainly lines up with stop signs, fire, and even blood as being things human beings are biologically preconditioned to view as dangerous, However, "biological human" association of color breaks down quite quickly when analyzing a variety of films when it becomes clear that different films use color to have different meanings. "Raise the Red Lantern" uses red oppressively, while "Her" uses a desaturated red (pink) for loneliness, while using a bluish shade for closeness and warmth. Name any color association that is considered "inherent," and countless counter examples can be found.

The truth is that the filmmaker creates the meaning of the color as they work. During the exposition of a film, in addition to introducing audiences to the characters and setting, the filmmaker has the opportunity to introduce how color will be used narratively. It is likely that, absent those meanings being created, that overall cultural bias will come into play for the audience. But the filmmaker always has the power to change those associations, and it's up to the filmmaking team to

clearly establish those associations in the opening of the film so that they can best be utilized through the storytelling process.

This is seen in countless productions across both film and television. One famous example is “Breaking Bad,”, where the colors are so prominent as to show up in the characters names (Jesse Pinkman, Walter White). The color of wardrobe is designed to symbolize character change. Supporting character Marie Schrader, who grows the least, consistently wears purple throughout the show. Walter White transitions from sickly greens and yellows (best exemplified by his puke green Pontiac Aztek) to movie villain black (including a “black hat”) by the end of the show. These sort of color decisions take advantage of audience’s ability to subconsciously track color, and work in conversation with cultural assumptions about color and character.

It’s important as filmmakers to understand the physiological underpinnings of color perception, and the power that gives us to implement our own meaning, as we plan and execute our productions.

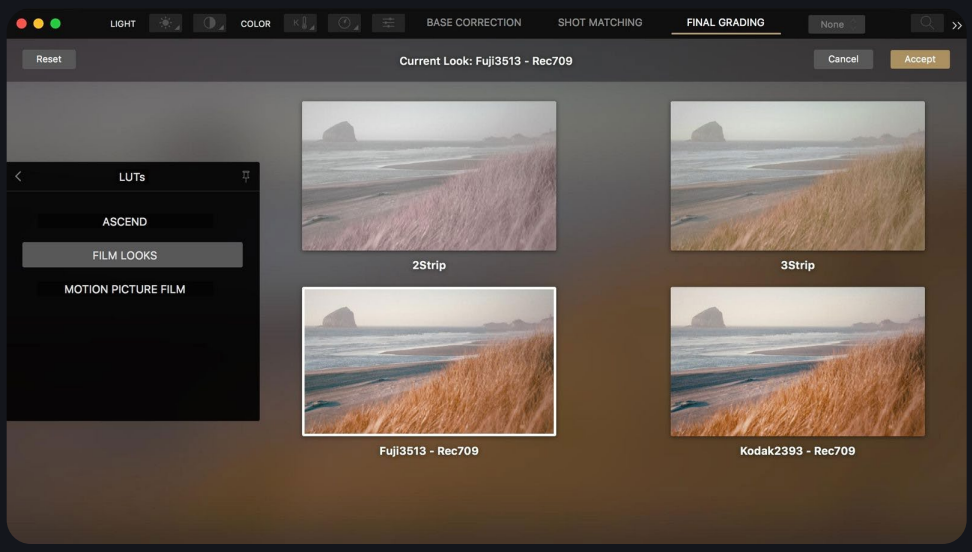
Contributed by Charles Haine, Filmmaker, writer, host and professor.

But for the same reasons that good color correction can enhance your story, bad color correction can radically degrade it. It is critical that your production team makes creative decisions about color early and communicates those decisions as the film moves down the pipeline.

# The Color Correction Process

As mentioned earlier, the DP will craft LUTs and CDLs during test shooting. These files assign an initial look to the footage, which gives each team a general idea of what the DP intended for color, brightness, and contrast. But they only serve as a rough blueprint for the color correction process. The colorist will build a brand new set of looks for the original camera files. This ensures the artistic intent of the DP is achieved, while also addressing the technical considerations of each shot individually.

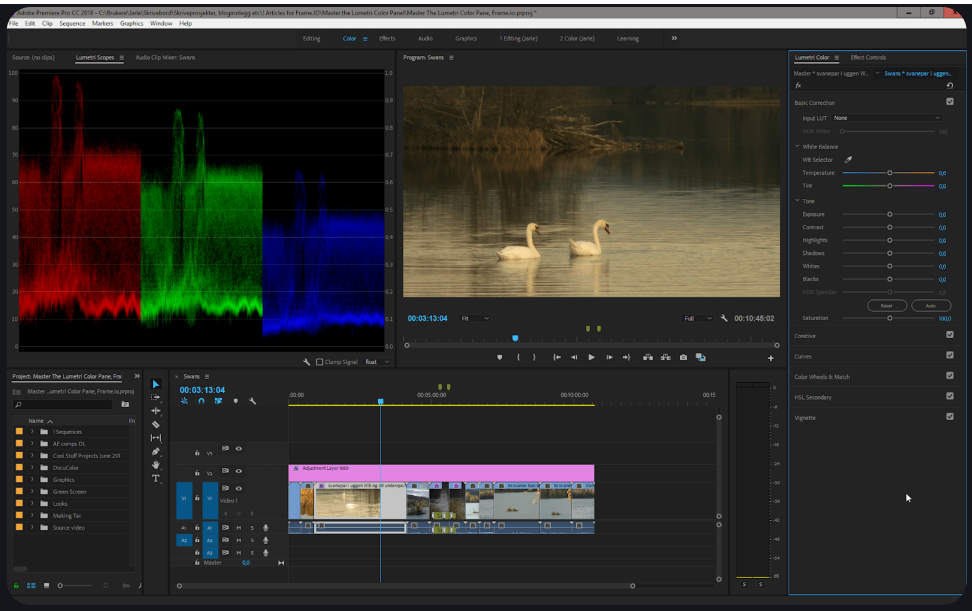
Apple ProRes Raw footage.



Color correction is usually handled by a specialist team. After picture lock and conform, EDLs, XMLs, or AAFs of the final edited sequence are sent to the colorist, along with the original camera files. Once the film is shot-matched and given its artistic look (with final approval by the DP), the sequence is prepped for the last steps of post-production.

[Note: Though color correction is usually the final step in a feature film's workflow before delivery, it is sometimes necessary to be a bit flexible. Because VFX work can often extend very close to the project deadline, color correction may take place gradually as individual shots are finalized and handed off to the coloring team. It all depends on the schedule.]

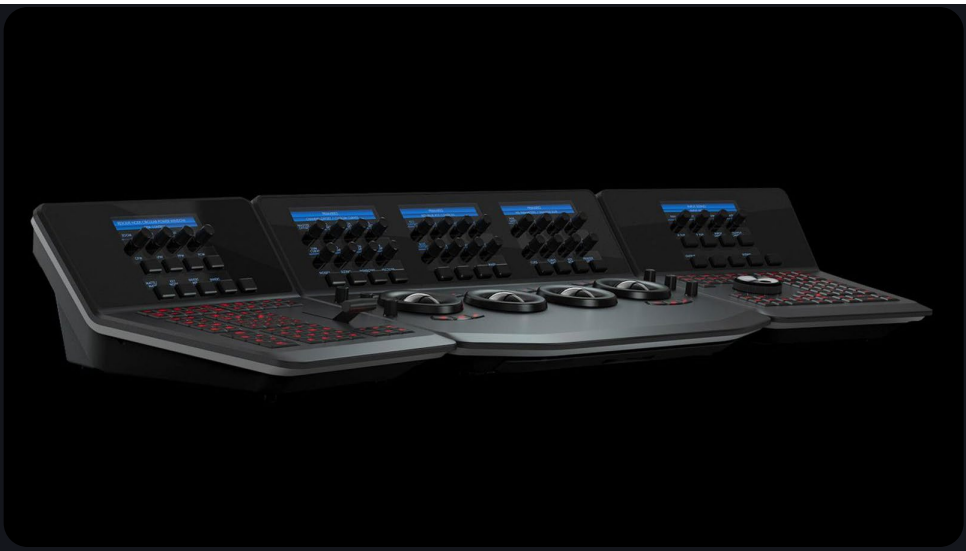
Of course, not all projects require the precision and cost of a dedicated color correction team. If this is the case for your project, your editorial team might be able to handle the coloring process inside their NLE. This decreases workflow complexity since they will not have to pass on the sequence after conform. Even if they use another program to color correct the footage, like Resolve, these will only be internal handoffs on the same machine/network, rather than to an external color team which is probably in a completely different facility.




As with everything in the post-production process, you will need to balance resources. Top of the line color correction brings a level of precision that is difficult to match, but it is expensive both in money and time. Alternatively, NLE-based color correction is much faster and cheaper but lacks the finesse of dedicated color hardware and software.

A professional colorist will also come with an extensive hardware setup that allows greater precision and speed when performing the corrections and also high-accuracy monitoring.

All of the colorist's tools can be used with a keyboard and mouse, but a dedicated control surface allows the colorist to use both hands to control multiple values simultaneously, jump quickly between tools, and do all of that without taking their eyes off of the image, which is very important.



## Color Correction Hardware

Professional color correction suites are an almost mythical domain to many who peek in from beyond their secluded walls. Huge DI grading theaters seem like the ultimate luxury, with enormous displays, cushioned chairs, perfect mood lighting, and oh so many buttons and knobs. These high-dollar tools look the part, and if you're passing off your project to a dedicated colorist, they probably fill you with confidence (otherwise you wouldn't be paying for it).

Newer filmmakers might think that such tools are beyond the reach of mere mortals. It is possible, however, to achieve

quality color correction on a reasonable budget. If you are handling the color work of your project in house, you should consider building out a dedicated workspace for this task.

Of course, filmmakers planning to build such a workspace need to learn how to use the necessary hardware tools, but they must understand the basic requirements of a color-focused working environment. Such considerations will help maximize the results of your color work.

## The Suite

The first thing to understand is that the color suite is a specialized, highly-controlled environment built for a very specific purpose—to work with color as efficiently and precisely as possible. This purpose brings with it special requirements that will not make as much difference for other parts of a post-production workflow. That means that building out a color correction suite is a significant investment that really only fulfills a single function. Before you get started, you need to determine if such an investment will pay off for your project/workflow.

Assuming that it does, the first task is to identify your actual workspace. Color grading suites can be almost any size or shape, but your post-production context will dictate exactly what you need. If you're planning on having 10 or more clients in the room to review color, then you will need a large space akin to a small theater. But if you only plan on having 1 or 2 reviewers in the suite at any given time, then a nominally-sized office should do just fine.

There is also the option of setting up your suite for remote color correction, where clients are not actually in the room with you, but are at another location following along. This is a more advanced type of suite, but can be tremendously useful

for workflow that stretch across the country or the globe.

Remote Color Correction in Resolve

Contributed by Charles Haine, Filmmaker, writer, host and professor.

## Light

Once you have your space selected (or constructed), your next task is to control all ambient outdoor light. This control is the first, fundamental step in creating a color correction suite. Any stray light that enters the suite is a problem. Even if it doesn't hit your screen, ambient light influences your perception of color, because it changes throughout the day. That means your eyes and brain will naturally adjust without you knowing it, which will influence how you see the colors in footage.

To gain control over ambient light, many color correction suites are located in the interior of buildings, with no windows to the outside. This prevents any sunlight from leaking into the room, with is great for ensuring consistency in your work. However, this lack of natural light exposure can have a negative emotional/physiological effects if you are working in it for most hours of the day, so be sure to still go outside every once in a while. Should you choose to use a room with windows, make sure you have substantial blackout curtains to block out as much sunlight as possible.

Once you have complete control over ambient light in your suite, it's time to add your own. Every bulb in your suite should be balanced to a 6500k color temperature. This balance corresponds to the D65 white point of your computer monitor, which will prevent your eyes from adjusting to an off-colored light source. You also need to make absolutely sure

the bulbs have a high Color Rendering Index (CRI), of at least 90. The higher the CRI of a lightsource (up to 100), the more accurate colors will appear under their illumination. In general, it's a good idea to make sure all the bulbs/fixtures in your suite are the same model, to eliminate any inconsistencies in your light source.

When it comes to light placement, it really comes down to preference. The only hard requirement is to ensure none of your lighting pollutes your picture. Other than that, you can distribute them around the suite as needed, to make sure other tasks (like walking around or reading) are not impeded. Dimming is a useful feature, but you will need to buy lights with this capability (not all CFLs and LEDs can dim). On top of that, make sure the dimming function is high-quality, and does not produce a visual flicker or audible buzz, both of which can seriously degrade a working experience over long hours. Some lights will even change color when dimmed.

## Paint

The final step in building your color correction suite is to establish a neutral color environment. How do you do that? By painting the walls neutral gray, specifically *18% gray* or *middle gray*. The name of this color is derived from it's position on CIE lightness scale, roughly halfway between absolute black and white. The "18%" refers to the amount of visible light this color reflects. The reason this gray is necessary is that it keeps your perception of color constant. If your walls were bright yellow, your eyes would compensate somewhat without you knowing, and you wouldn't see yellow in your footage quite as much. That might mean your actors come out looking a bit like bananas, which your client might not appreciate. So, gray walls serve as a neutral point on which your eyes can establish a baseline of color and brightness.

This paint can in some cases be mixed at a local store, but the

results are not always consistent or accurate. The best option is to order the paint from a specialized manufacturer. Keep in mind this paint can be very expensive (~$80 per bucket), but the results are difficult to match. However you source your paint, get a sample and test it in your suite. Once you have selected a paint, any wall that is in your view (including your peripheral vision) while you work needs a coat.

Once all of this is done, your suite will be a conducive environment for color correction work. Now, you need to fill it.

## The Screen

The first thing you will need in your suite is a viewing device. The screen, monitor, or projector you use is the digital window into your creative world of color. Just like real life windows, some are clearer than others. If you want to see your color work as clearly as possible, you need to make sure your window is as good as possible. There's really only one way to keep this digital window clear: calibration.

### Calibration

Serious color correction requires proper calibration. Without it, the visual style of your film might not turn out the way you want it. A character's skin might look sickly, or a scene might cause an unintended emotional response from audiences. But a lack of calibration will cause issues before the project is even finished. If a shot is viewed on seperate, uncalibrated machines, there's not really a way to know which is displaying the image correctly, so it's impossible to make creative or technical decisions.

As we covered earlier, the colors that your computer outputs are just digital values. It sends them over a cable to your

screen, which then reads those values and does its best to produce them. But your computer can't really "see" the colors that your display is outputting, so you might perceive a different color from the computer is trying to show.

Calibration fixes this issue by cross-checking the display's color output against known values. If a color value is being displayed incorrectly, the computer will be able to detect it and adjust the color settings of the monitor to compensate.

The most common color calibration tool is known as a *probe*. These are usually small hardware devices that connect to your computer, and then attach to or rest on the surface of your screen. The computer will run a test pattern (usually included in the probe's accompanying software) that the probe measures as it is displayed on the screen. As the results are analyzed, the color calibration software will adjust the gamma and color gamut output within the monitor's color profile. Some color calibration software allows the user to choose the color standard, like sRGB, Adobe RGB, or DCI-P3.

Some of the most popular external color calibration probes are manufactured by X-Rite and Spyder. These probes are relatively cheap, so they should fit into most any budget. Even if they were several times more expensive, they would probably still be worth the investment. You will often see monitors advertised as being factory-calibrated, which can be helpful, but it's still a good idea to invest in a calibration device. Monitors drift over time, so it's essential you have a way to re-calibrate them. However, this doesn't mean you always need an external calibrator. Some high-end professional displays have integrated calibration sensors. These units are pricey, but for certain workflows they may be worth it.

## Display technology

Now, just because a monitor has been calibrated does not necessarily mean it is the best choice for your color work. There are many other factors to consider. Fortunately, display technology keeps advancing at a tremendous rate, yet prices continue to fall. There are a few technical specifications that you should look for in a display for your suite.

First, you will need a display with a suitable resolution. In general, you should aim to display your footage on a screen of the same resolution. If you try to grade 4k footage on a 1080p screen, you will have to fight against downsampling artefacts. And if you try it the other way (HD footage on a UHD screen), you that can introduce other issues. This is because the methods for upscaling resolution differ across display equipment, and so things may not always look exactly the way they should. If you do need to color footage that is lower resolution than your screen, it is a good idea to only display it at 100% resolution without upscaling. It is best to select a screen that matches the resolution of your most common footage.

Other than resolution, you need to consider contrast ratio. More is generally better, and you should aim for a display with as high of a contrast ratio as you can afford. For serious color work, 1000:1 is widely considered to be a minimum. However, if you plan on working with HDR content for an HDR delivery, that minimum will increase drastically.

Along the same lines, you should aim for a display with the deepest black levels and widest color gamut that your budget allows. In the past, plasma screens produced the darkest, richest images, but this technology has become antiquated. The newest high-performance screens use OLED technology. These displays are expensive, but their visual capabilities are remarkable. That said, there are tradeoffs to using an OLED screen, like image burin-in. Not only that, but new technology is entering the display market every year, so it probably won't be long until there is something better.

The final technical specification you should aim for is a 10-bit display This will give you a lot of flexibility in working with your footage, as your display will be able to produce many more colors than more common 8-bit panels (over a billion versus ~16 million). There are also 12-bit displays available, but they are hugely expensive, and require additional, complex equipment and adherence to technical standards that few workflows can handle.

Beyond these broad specification, you will have to choose the actual type of display that fits your workflow needs. If you're grading for cinema, you'll want to grade on a projector. But if you're grading for web, you can just use a computer monitor. It's always a good idea to see what your image will look like at its intended destination. That will help you make sure everything turns out right. For example, if your project is headed for broadcast, then you should have a TV in your suite that is conformed to Rec.709. This will give you the most accurate representation of how viewers will encounter your work.

The same concept applies to bit depth. If you know that your content will only ever be viewed on YouTube (which will definitely be 8-bit), then it's actually best to monitor it in 8-bit.

Once you have a display in your suite, there are a couple of ergonomic factors that will influence your color correction. The first is your display's brightness. In order to conform to SMPTE standards for a color suite, you should set your monitor to about 25% brightness (as measured by a light meter with the screen showing pure white). This level of brightness will be just enough to see your immediate surroundings in the suite, but low enough that your screen doesn't produce too much ambient light.

Depending on the size of your suite, and the distance you plan to sit from the screen, you will have to decide on the size

of your display. A common rule of thumb is to measure your eyes' distance from a screen's position, and then divide that measurement by 3. This number is the vertical height of the monitor that is recommended for that distance. For example, if you sit ~4 feet from your display, then a 30 inch display is recommended. In theory, this will give you a full view of the image, but without too much eye strain. However, these are just guidelines. Your exact ergonomic preferences should be the deciding factor.

Following these recommendations will give you the best results for viewing your images. The final step is to equip your suite to manipulate that image.

## The Control Surface

Any professional colorist will tell you that the fastest, most efficient way to manipulate an image is with a hardware control surface. A serious color correction suite will feel incomplete without one. While you can certainly achieve excellent results using just software controls, once you've made the effort to set up a dedicated suite, there's really no reason to skimp on this final investment.

That said, control surfaces can be a big investment. DaVinci's full control panel costs $30,000. And there are actually even more expensive options. But you don't have to spend a new-car's-worth on color control hardware. Over the last decade, a number of affordable systems have come to market, with some costing only a tenth of DaVinci's full board. There are even some options (with fewer features) that only cost a few hundred dollars. Of course, the tradeoff is between price and capability. More expensive tools give you more creative and technical control.

You might be wondering what these capabilities are. Well, to

start with, a control surface controls the tools of digital color correction–brightness, hue, saturation, tonal ranges, secondary qualifiers, and vignettes. Pretty much any parameter of an image can be adjusted with a control surface. That's why there are so many buttons, dials, knobs, and trackballs. Each one does a specific thing. And this is no coincidence. The software tools we use for color evolved to mirror the hardware controls. These tools were originally made to be manipulated by hand, but as software advanced, they were adapted for the mouse and keyboard.

So, naturally given the origin of these tools, it's easy to see why color correction benefits from a physical toolset. Tactile controls allow for intuitive operation, and helps the user develop precise muscle memory. Of course, if you are used to a hardware control surface with trackballs, then software color wheels will make perfect sense to you. But, you won't be able to work with them as quickly simply because a mouse is not a trackball.

The main benefit of a hardware control surface is speed. Instead of making individual adjustments one at a time, you can adjust as many settings as your fingers can reach. When multiple corrections interact together, you can play with them and explore their influence at the same time. With your hands resting over two color wheels on the board, you can adjust brightness, saturation, and hue for two separate tonal ranges are the same time. Such a feat is not possible with a mouse. And with enough practice, you can make almost any color adjustment without ever taking your eyes off the image. That saves time, which saves money. So, depending on how often you are color-correcting footage, a control surface could have a strong return on investment.

But we also need to consider the soft-return of such an investment. Whether it's right or not, clients now associate control surfaces with serious color work. If they see you using one (and using it well), they will perceive you to be as skilled

artist worth their repeat business. This is not to say that you can't keep clients if they see you use software tools (obviously that's not the case), but many clients will have a higher regard for tools they are not familiar with. After all, those mechanical buttons and switches look very fancy to people who don't understand what they do.

At the end of the day, a control surface is just another tool. If you know how to use it, you will be able to color your project faster and more accurately. But, it's an advanced tool that will only really add value if your workflow (and skillset) can utilize it.

Contributed by Denver Riddle, President at Color Grading Central.

Whichever path you choose for your project, the color correction process will play a tremendous role in shaping your film, technically and artistically. For best results, you should plan your options for color correction based on the scale of the project, the complexity of your distribution channels, and the creative requirements of the story.

## Online/Finishing

The online edit, or "finishing" is usually the final step before the film is exported out into its final format as a single video file. Once the colorist has finished matching shots and crafting the look of each scene, the project moves on to the finishing artist (sometimes called a finishing editor or online editor).

Up until this point, the film has been in the hands of a series of

specialists who focus entirely on one task. The editor crafts the story. The colorist builds the look. The VFX house handles big, complex shots, but smaller things are left in the edit.

The finishing artist is a generalist who can do a little bit of everything –edit, color, and VFX. The finishing artist is an image-quality fanatic who ensures that everything looks perfect before it is sent off to the theater or the airwaves.

## VFX

While any complex VFX shots are usually sent off to a VFX house, smaller effects may be left for the online editor.

If the editors decide to combine two performances in the same take, they will probably be able to do a pretty-good job in their NLE, but the finishing artist will recreate that shot with their more advanced tools. The same goes for other simple VFX tasks like removing a sign from a wall, or replacing a sky.

If the editors have used any complex cross-fades or speed-ramps, the online editor will also recreate those.

## Titles

Unless an external company was hired to produce the titles, the online editor often handles that part of the workflow. It's common for the editorial team to work with the director to place the titles and time them properly, but the editorial team usually doesn't put the full polish on them.

While all NLEs can create titles, they don't have the level of nuance that a finishing tool provides. Editorial teams also often work with a

lower-resolution version of the film, and so it would be pointless for them to spend hours perfecting the titles if they can't view them at their full resolution.

## Last-minute changes

Because the online editor is the last person to touch the project, they are often asked to handle last-minute changes to the color, or even to the edit.

If the client or the producers decide at the last minute that they've changed their mind about the color in one shot, it may be too late or too expensive to send the film back to the colorist. In that case, they'll ask the online editor to make the change.

Or, perhaps the VFX house just delivered a new version of an animation, and it needs to go in and replace the old version in the edit.

## Finishing Tools

Because the finishing artist needs to be able to do a bit of everything, a finishing tool must provide a full set of color, editing, VFX, and animation tools.

On the lower end, and even on some high-end films, After Effects is common for finishing because of its low price tag but broad feature set. Davinci Resolve is another affordable option which, with the addition of Fusion, is now a capable finishing tool as well as color grading application.

On the higher end, Avid Symphony, Autodesk Flame/Lustre, and Assimilate Scratch are all common finishing options. These options

are more expensive than After Effects and Resolve, but they are superior in capabilities.

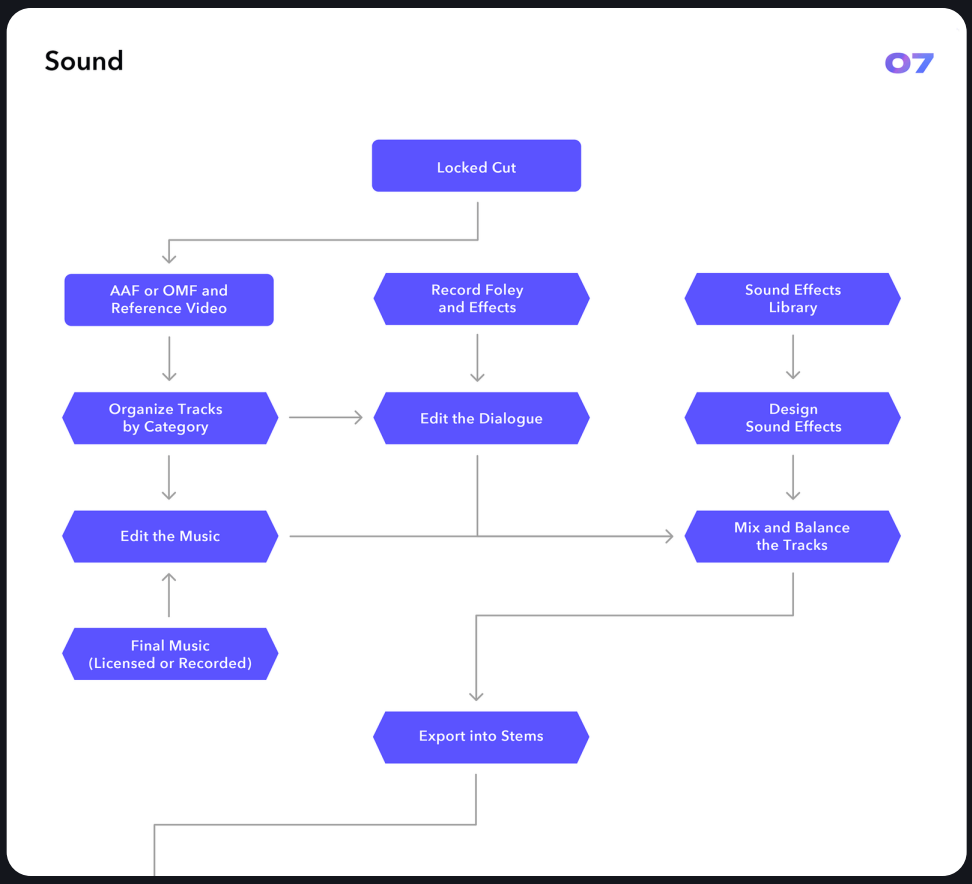
In some cases, especially on smaller projects, the finishing artist and the colorist may be the same person. And on very small projects, the finishing artist, colorist, and editor may all be the same person! Even in those cases, however, it's still common to separate the different processes. The editor will edit using an offline codec in an NLE and then conform and transfer the project to another tool for the color and finishing work, so the film will progress in stages.

CHAPTER 07

# Sound

Sound is the invisible half of the film. The sound team will record new effects, polish what's already there, and mix it all together.

*14 minute read*




This Chapter was contributed by the awesome team at Defacto Sound.

Specialized audio post-production work adds a of lot polish to a project. This part of the workflow can heighten the excitement, intimacy, immersion, and atmosphere of the story. It also ensures the content meets the technical specs of whatever platform it will be delivered on, while sounding its best in any given playback environment. In order to transform the initial materials to the final, polished mix, we need to follow a series of steps:

## Materials Needed for Post Sound

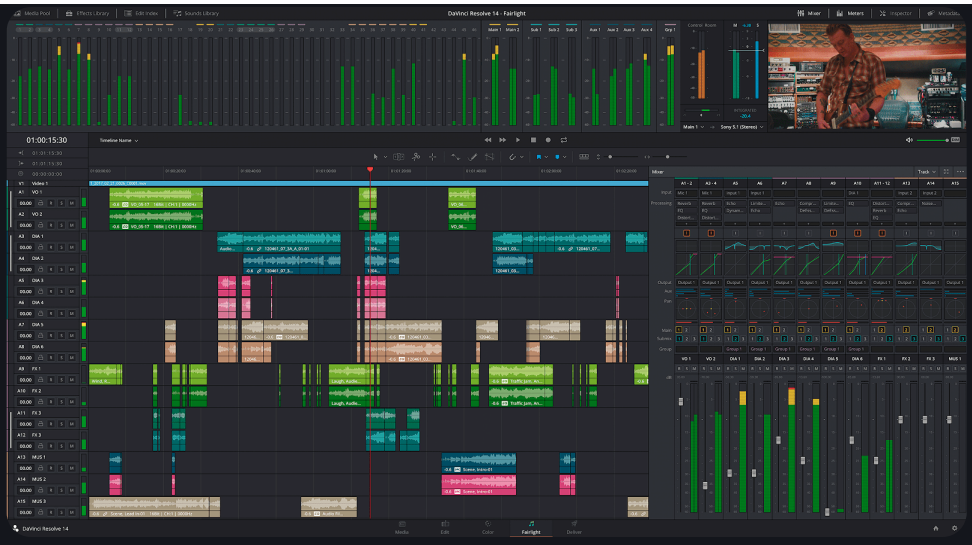
Most projects require an OMF/AAF, a quicktime export of the video, and a list of required specs and deliverables. A sound report from the production audio is nice to have if there are multiple mic choices, but not always essential.

## What's an OMF/AAF?

OMF and AAF files are special formats used for transferring projects between different editing programs. They allow post audio facilities to import audio clips from the timeline of your video editing software. It maintains the clips' relative positioning and fades, along with a few other features, like: panning, volume automation, and sometimes audio processing. Essentially, it is a snapshot of the project in your editing software that can be imported into Digital Audio Workstations (DAWs) like Pro Tools.

## OMF vs AAF

Both file types accomplish a similar task – exporting a timeline from one piece of software so it can be imported into another. AAF is a newer format that contains some additional metadata over OMFs (like track names), but not all video software can export AAFs. Regardless, both are completely acceptable ways to transfer materials to post audio.

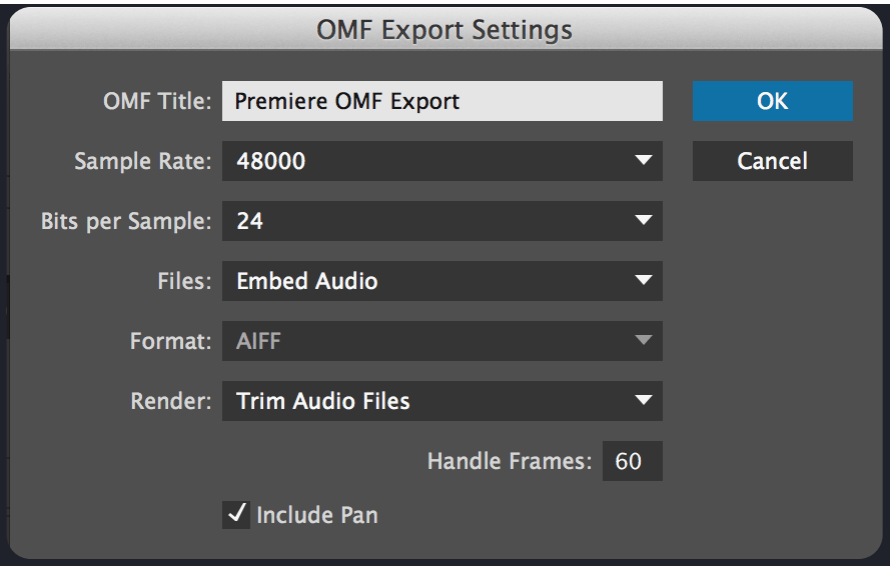


# Export Settings

Exporting OMFs/AAFs is different for every program. Below are program-specific guides for some of the most widely used editing software.

## How to export from Premiere

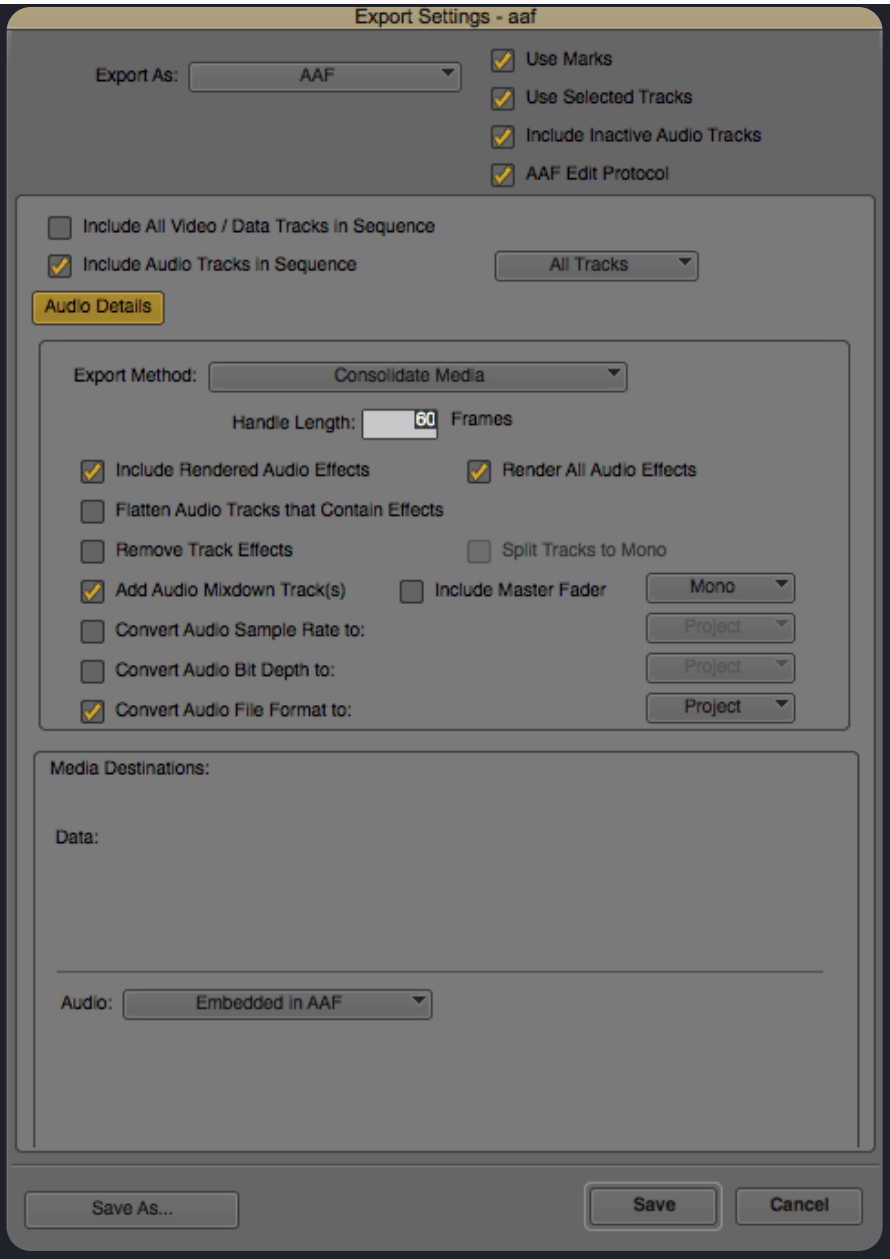
1. Select the tracks in your timeline that you would like to include in the OMF
2. Go to File – Export – OMF
3. You'll see the following dialogue box with the correct export settings.




Once you've exported the OMF, you are good to send!

## How to export from Avid Media Composer

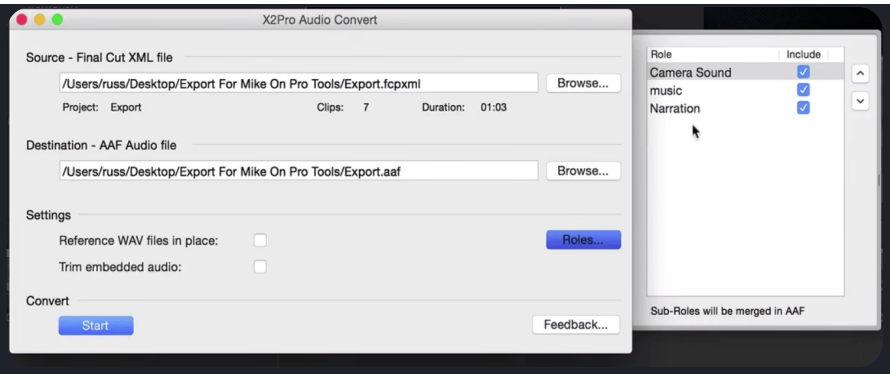
1. Select the tracks in your timeline that you would like to include in the AAF
2. Go to File> Export >AFF
3. Match the settings below in the export settings.



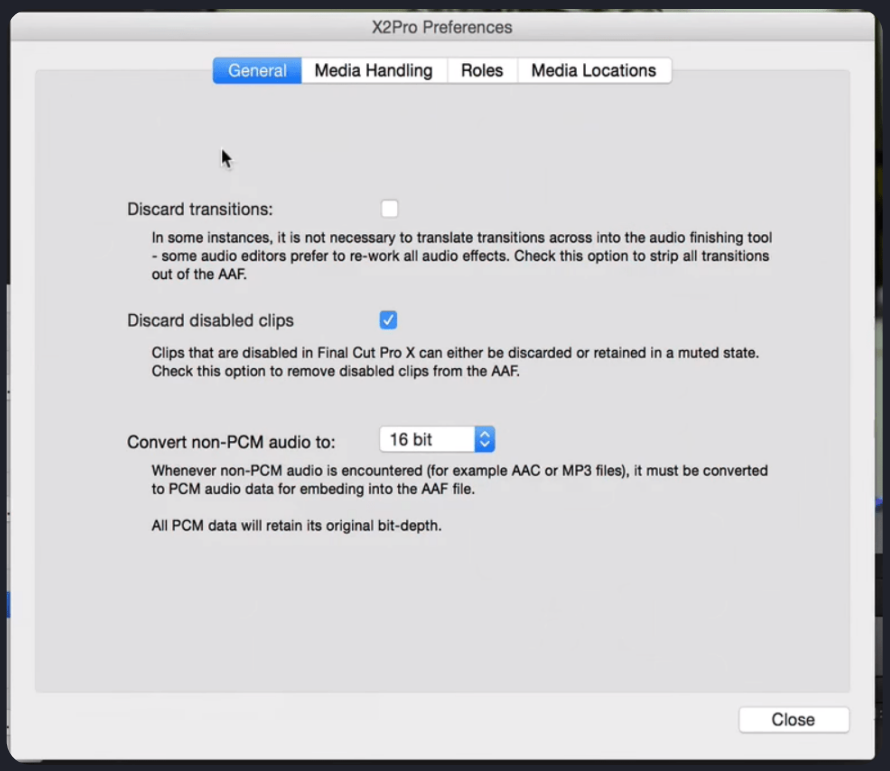

Once you've exported the AAF, you are good to send!

How to export from Final Cut Pro X

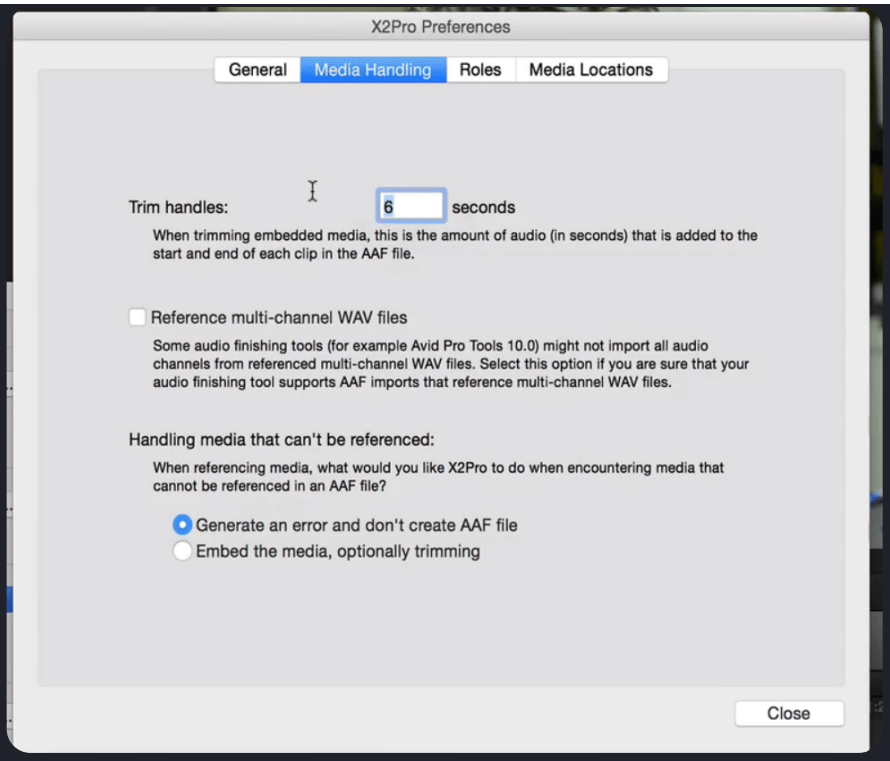
Click on the Roles Icon and select what you would like to include in the AAF.



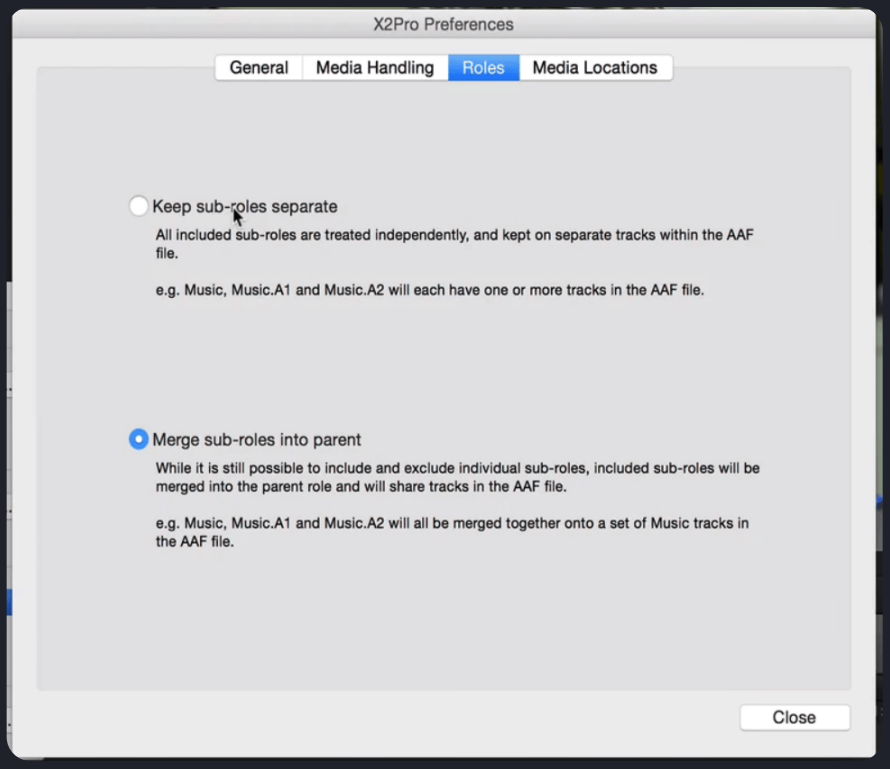

The General tab in the preferences menu should match these settings.



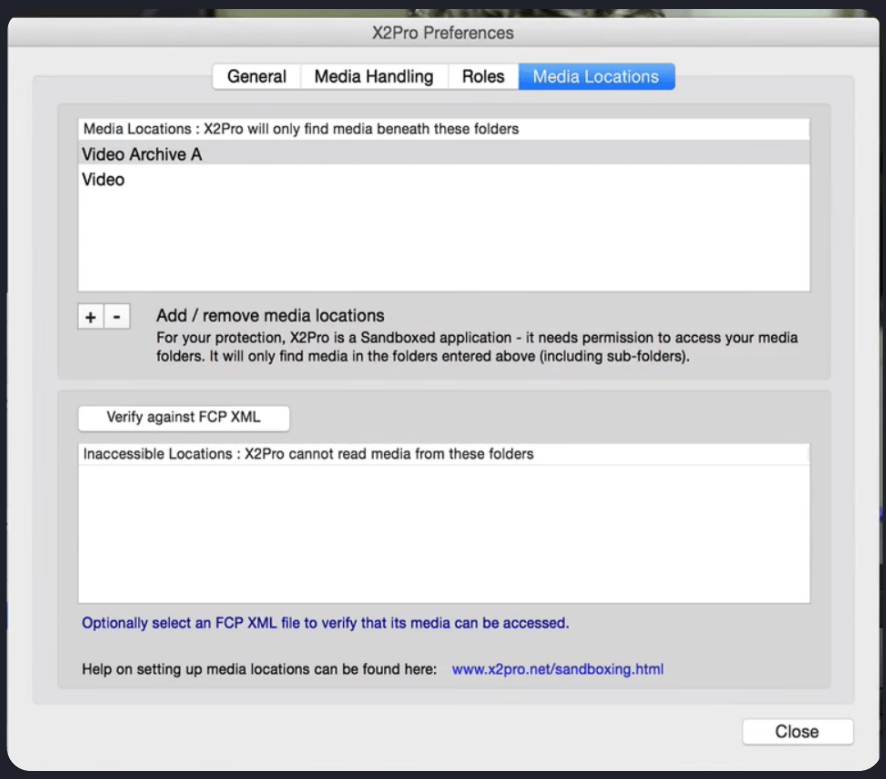

The Media Handling tab should match these settings.




Select "Merge sub-roles into parent" on the Roles tab




Under the Media Locations tab you can select the folder in which all of your audio assets live, so that x2pro may reference them correctly.




## Video Export

Here is a list of preferred video codecs for Pro Tools 11 and higher (as outlined from the Avid Knowledge Base):

- DNxHD (Preferred)
- Apple ProRes 422 (including HQ, LT and Proxy)
- DV25
- DV50
- H.264 (Performance may vary)

720p and 1080p resolution should be sufficient for an audio post facility. This allows the operator to save processing power for audio.

1080p still provides enough detail for your sound designer, and the smaller files will speed up exporting and uploading / downloading.

## "Spec" Requirements

Every project has its own special requirements. Here's a list of relevant info to pass along to post audio to ensure the final mix meets those requirements:

- How the content will be distributed (broadcast television, web, film festivals, specialized installation, etc.)
- What formats are needed for the mix (Stereo, 5.1 Surround, 7.1 Surround, Dolby Atmos, etc.)
- Any other relevant audio specifications (in the case of broadcast television, these specs will usually be contained in a guide from whichever network the spot/episode will be airing on)

# Post Audio Workflow

## Basic Step by Step Process

1. Track Management
2. Dialog Edit
3. Music Edit (if needed)
4. Rough Mix (variable)
5. Sound Design/Sound Effects Edit

6 Final Mix

7 Review and Revisions

8 Deliverables

# Track Management

The first step is organizing the audio clips from the AAF/OMF in a digital audio workstation. It's important for each audio clip to be organized on a track with similar audio clips before the post audio process can continue (dialogue with dialogue, sound effects with sound effects, etc.)

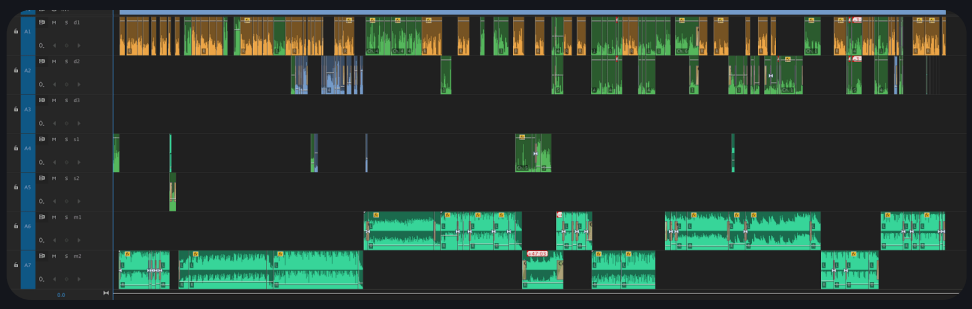
To make this process as quick and smooth as possible, the tracks in the editing software should be labelled and organized before the AAF/OMF is created. It is preferable to send a larger number of well organized tracks, instead of combining different kinds of clips together on a smaller number of tracks.

Categories to organize audio tracks into:

- VO (Voice Over)
- Dialog
- Production Effects (anything recorded from set other than principal dialog)
- Sound Effects (separate from Production Effects, anything that was added in post)
- Music

If you have a mixture of mono and stereo files in your edit, it's

helpful to keep these organized as well. For instance, all mono sound effects should be placed together on tracks above your stereo sound effects.



A special note about dialog tracks – it is best to keep all of the different microphone channels from production vertically grouped together. That way, your dialog editor can easily choose the best microphone for each scene. A good minimal track layout includes a boom microphone and a lavalier microphone. An expanded track layout would include Mix Left, Mix Right, Boom, Lav 1, Lav 2, etc.

## Dialog Edit

The dialog edit is an often-overlooked but extremely important step of the post audio process. When it's done right, the dialog track should sound completely natural and unedited.

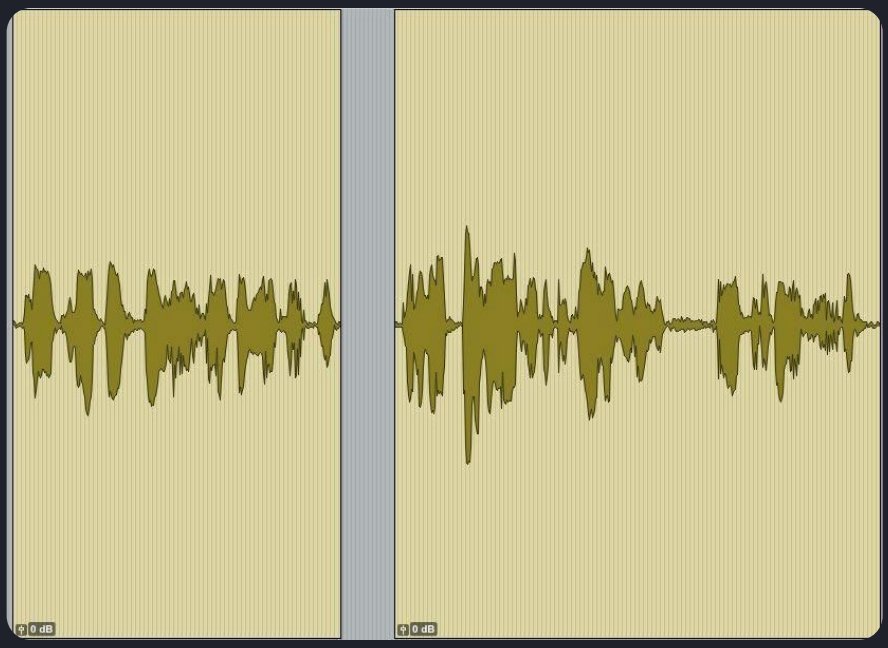
The Dialog edit is usually the first task after track management because it provides a crucial anchor in many projects. Here's a quick overview the process:

- Find the cleanest room tone possible from each mic in the scene.
- Use this room tone to bridge dialog bites together and add air for longer fade ins/outs (especially important if the room tone is particularly noisy or loud).
- Smooth out any chopped words or breaths.
-

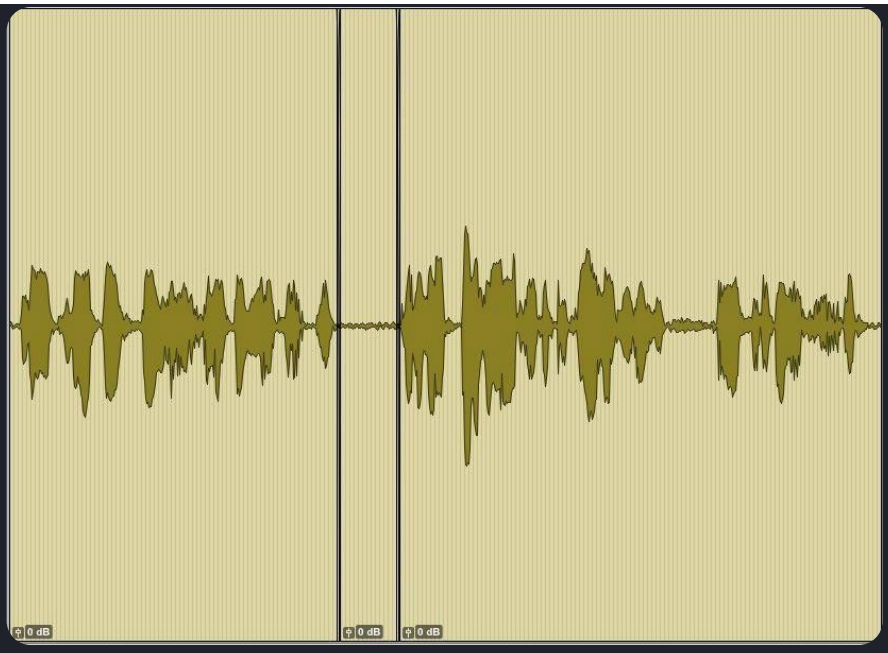
- Smooth out any “frankenbites” (Sometimes a word or sentence must be created if rerecording isn’t an option. This is never ideal but is sometimes necessary).

## Room Tone

The next step to a clean dialog edit is finding room tone for each of the dialog recordings. As mentioned, room tone is the ambient sound of the environment where a recording was made. When a dialog recording is cut up during the picture edit, gaps are often formed between lines (as shown in the screenshot below):




As a result, the ambient noise from that recording pops in and out with the dialog, often to a noticeable degree. Using clean room tone, you can fill those gaps that were created during the edit, creating a consistent bed of noise throughout that piece of dialog:




What makes Room Tone "clean"?

Clean room tone contains the ambient sound of a recording space and nothing else. Cloth rustling, distant car horns, and breaths are common things to avoid in your room tone. For example, if your chosen room tone has a car horn in the background, that horn will be present every time you use that room tone to fill a gap in the edit.

A desirable clip of room tone should be decently long so that it's easy to work with (depending on your exact uses, anything under 10 frames or so can be problematic). For instance, if you're filling a three second gap between two dialog bites, looping a 5 frame long bite of room tone can be tedious, and may sound obviously looped to the listener.

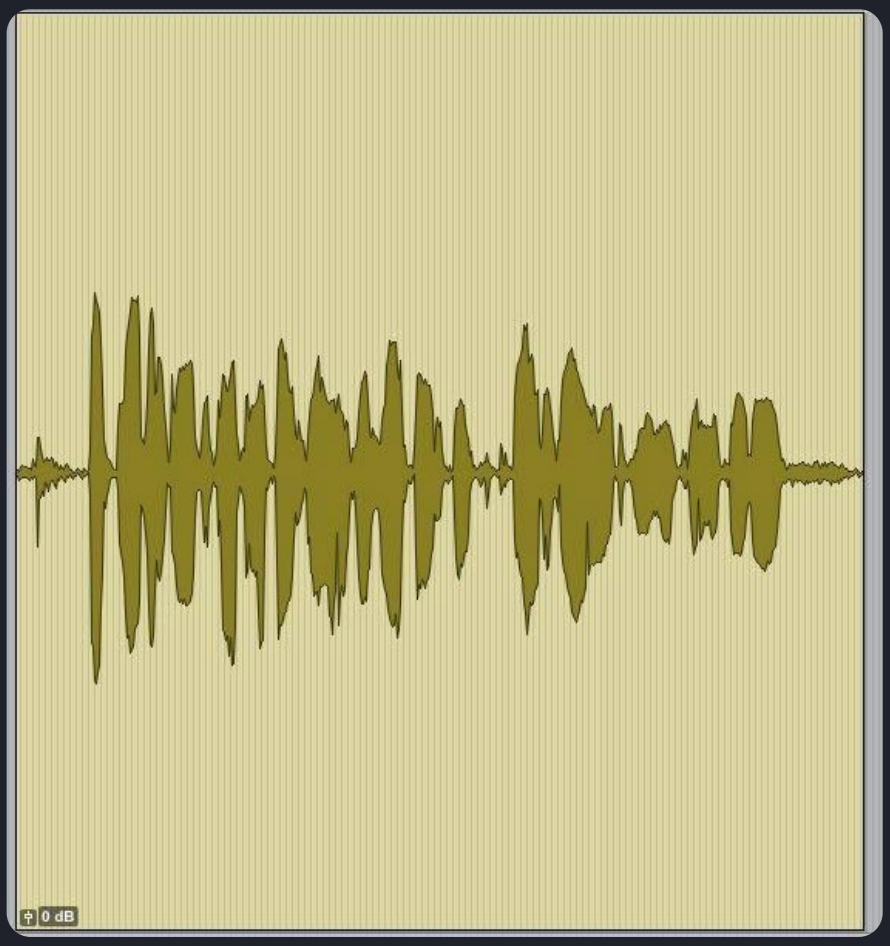
There are multiple methods of finding clean room tone. Here we list the most common ways, generally from most desirable to least:

- Room tone recorded by production sound – Sometimes production sound will intentionally record the ambient sound where a dialog recording took place. This is always good to

use when available. However, it is worth noting that even minor changes in mic placement can drastically change the sound of room tone. Also, over time, most ambiences will change slightly. It's always a good idea to do continuous listens to make sure the room tone continues to match a scene.

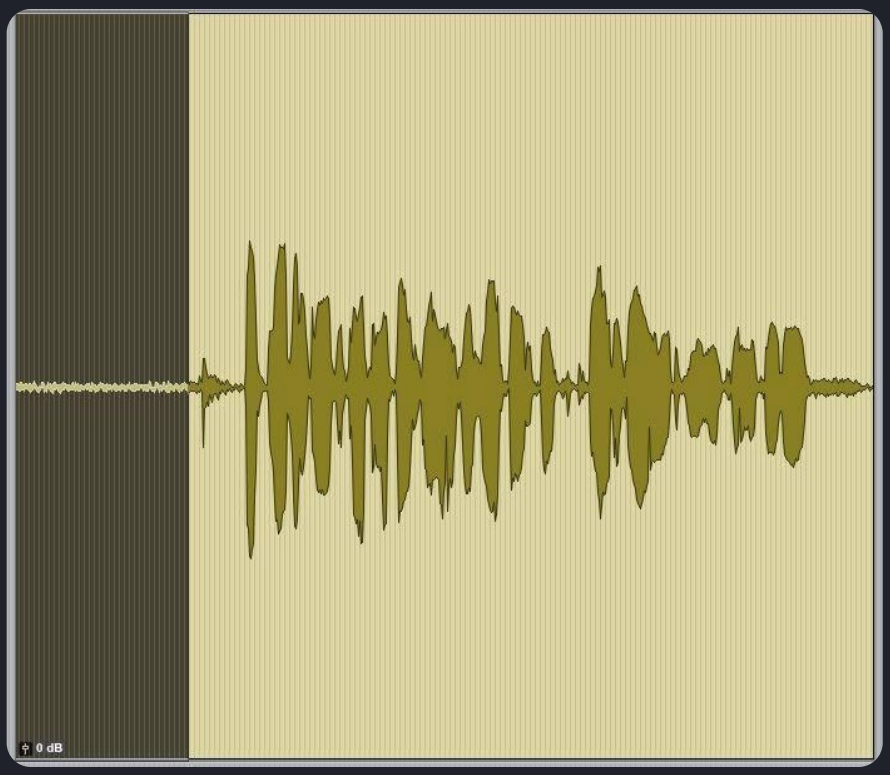
- Pulling out the handles of dialog bites – Clip handles refer to the extra recording before the start of a clip and after the end of a clip. The picture editor should provide handles for the dialog editor when exporting the OMF/AAF. This allows a dialog editor to pull out the edges of that clip to access the recording around that line.

For instance, say the following clip of dialog was provided in the OMF:




Since the picture editor included clip handles when exporting the OMF, the sound editor can pull out the start of the clip, as

shown in the highlighted area below:




The highlighted area appears to be clean room tone! An editor can copy that room tone and paste it wherever it's needed.

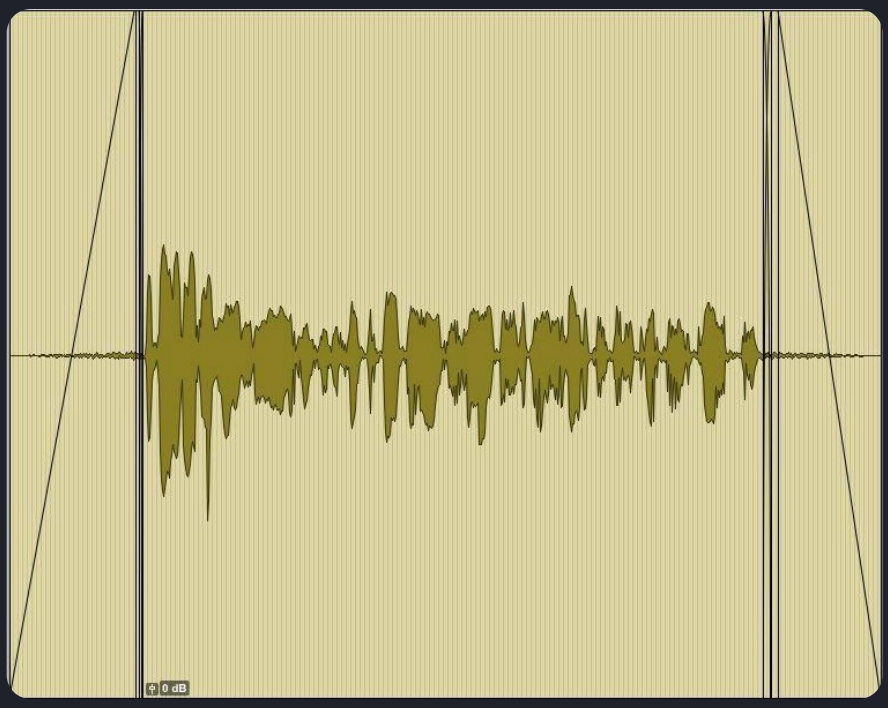
- *Splicing together room tone* – Oftentimes you won't be able to find a single stretch of clean room tone throughout the dialog recordings you're given. In these cases, you may have to take whatever bits of clean room tone you can and "splice" them together into a longer clip of useable room tone.

Other Uses of Room Tone

Room tone is used for more than just filling gaps between dialog bites. Here are someother common uses:

- Fading the background noise of a dialogue recording in/out – If you're dealing with a particularly "noisy" dialog recording (perhaps it was recorded next to an air conditioner) it can be jarring if that background noise pops in sharply when the

dialog starts. To smooth that out, we can use our clean room tone to create a gentle fade in/out of the noise:




- Covering up unwanted breaths/rustles/etc – Sometimes it's appropriate to cut the natural breaths out of a dialog bite. For instance, if that dialog is serving more as voice over (the speaker isn't on camera) maybe you want to remove those breaths. Instead of creating harsh cuts out/in to remove the breath, you can replace that breath with clean room tone. This same method can be used for any rustling or other unwanted noises between dialog in the recording.

Using room tone, you should be able to create a full dialog track with no noise popping in and out. Each dialog recording can play in and out naturally by creating fades with that room tone, and ultimately your dialog edits can sound seamless to the listener.

Frankenbites

Frankenbiting (also commonly referred to as franken-edits) is a phrase to describe a line that has been edited together from multiple pieces of dialog. If an interviewee didn't say the line you want, you may be able to cut together various parts of their recording to "manufacture" that line.

It's best to avoid Frankenbiting when possible, as it can be difficult to maintain a natural sounding delivery. That said, sometimes it's needed for the narrative goal of a project. Here are a few tips for creating clean Frankenbites:

- Avoid tonal shifts – One of the clearest examples of a messy Frankenbite is a big tonal shift in the middle of a line. Listeners will notice if a line delivered in a slow, low voice is suddenly spliced together with a fast, higher pitched delivery. Try to find bites that have a similar tonality and delivery when possible.

- Edit on consonants rather than vowels – There are some vocal sounds that are better suited to editing together than others. In general, vowel sounds are difficult to edit together in a natural way. Since vowel sounds feature more tonal content than consonant sounds, any shift in the tonality between the edited lines will be noticeable. Conversely, consonant sounds like esses , th's , and hard k's and t's are easier to cut together naturally. When possible, try to create edit points on consonant sounds rather than tonal, vowel sounds.

- Evaluate if the Frankenbite is worth an unnatural sounding line – While you may technically have the words needed to manufacture a particular line, there may be no reasonable way to make that manufactured line sound natural. If you find this is the case, it may be worth reevaluating how important it is to the overall project to have that line. If it's essential to your narrative goals, it may be worth having a brief bit of unnatural dialog. If it's just a small change, that frankebite may not be worth alerting your listeners to a funky edit. Depending on the importance, it may be worth contacting the original actor/VO artist to see if they can re-record the line.

The dialog edit has just as much to do with listening to the space in between the words as it does the words themselves. Ideally, the audience shouldn't be able to perceive any sudden changes in room tone or cadence. A poor dialog edit makes the post production process negatively visible to the viewer, through awkward jumps in the dialog's tone or cadence.

### How to Choose Dialog Recordings

If the dialog in your project was recorded with multiple microphones, the first step of the dialog edit is choosing which microphone recording to use. Here are a few questions to keep in mind when choosing:

- Which recording has the better "tone"? – Choose the microphone that offers the most realistic and full tone. When possible, it's good to avoid recordings that sound "tinny" (lacking low frequencies) or "muffled" (lacking high frequencies). Recordings can be enhanced after the fact by using an equalizer or other similar processing, but it's good to use the recording with the best tone to begin with.
- Is one recording noisier than the other? – Sometimes, one of the microphones will have picked up a lot more of the ambient background noise than the other. Some microphones (like lav mics) can also pick up clothing rustles if the speaker was moving around during the recording. In general, the recording with less noise is preferable.
- How "echoey" is the recording? – Depending on the microphones that were used and their placement, some recordings may sound more "roomy" than others. In general, using the closer sounding recording with less echo is desirable (especially for "talking head" style interviews). There are cases where the more roomy sounding dialog could be

appropriate (for instance, a narrative film where you want to recreate the natural sound of the space the dialog was recorded). You have to ask yourself if the dialog should sound "clean" or "natural" for your needs. A roomy sound is also less noticeable when the speaker is on screen. Voice-overs are typically as dry and clean sounding as possible.

There are times when no one microphone is superior in every measure. Use your best judgement for which recording provides the best quality for your particular project. This is also why it's helpful to send all available recordings to a sound studio. They may find it's easier to improve another mic than the one that was initially chosen.

In general, you want to use the same microphone recording for the entirety of a scene. That said, there may be times where one recording is superior to the other with the exception of one small bite (perhaps the lav mic had a big rustle over a particular line). Sometimes it's best to swap in a different recording for that one line, rather than commit to an overall less desirable recording because of one hiccup in the better recording.

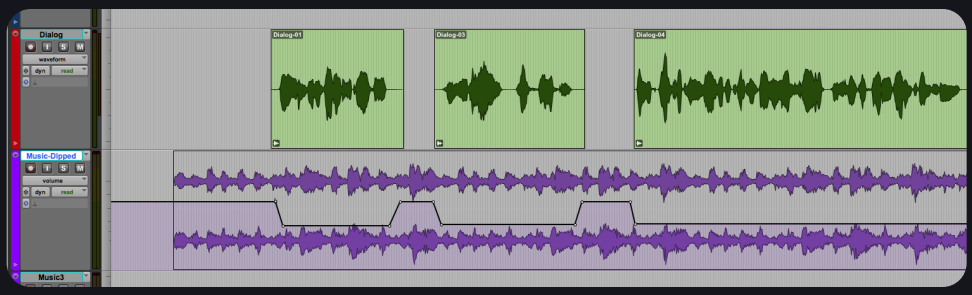
The ultimate goal of the dialog edit is to produce a seamless, natural sounding dialog track. Dialog recordings are often chopped up, spliced together, and spread out during the picture edit. This happens both for editorial and creative reasons. The dialog edit aims to smooth out those chopped up bites to make the edits "invisible" to the listener, and give the impression that what you hear is what was actually said.

## Music Edit

The music edit is the process of smoothing out how each song or

music cue will fade in, fade out, and transition. If music is specially composed for the project however, this step may simply be a check-up.

It is important to remember that picture is oftentimes more malleable than music. Music has a multitude of factors that make even a small edit noticable if not done correctly. Tempo, rhythm, instrument decay, and sustain are just a few factors that can throw off the pacing and groove of the music and the project as a whole.




Smoothing out music in the edit by dipping beneath dialog.

It is often better to make slight alterations to picture when lining up a music track with visual cues, rather than try to make a forced music edit work during post audio. Picture editors will often create a temporary music edit. So, many of these potential problems will be solved before audio post. In any case, keep the lines of communication open so your audio editor can collaborate if need be.

## Rough Mix

Once the dialog and music is edited, it's common to do a rough mix of these elements along with any production audio or sound effects added by the picture editor. This step can help the sound designer see and hear the project as the editor did, albeit a much cleaner version. This also helps to make informed story-driven sound design decisions.

## Sound Design

Sound design constitutes any elements post audio adds to enhance the soundtrack beyond the music and production audio. Sound design can be broken down into a few categories:

- Hard Effects – Sounds that are typically not recorded to picture (explosions, car engines, doors slamming)
- Foley – Sounds of the characters and the objects they touch (footsteps, cloth movement, prop effects)
- Backgrounds – Environmental ambiences (nature sounds, cityscapes, office murmurs)
- Emotional Effects* – Abstract effects out of the reality of the scene used to enhance emotional impact

*Emotional effects are the hardest to explain and are often easily confused with music. Many music compositions incorporate these types of effects as well (cymbal swells, eerie drones, etc). Non-literal sounds is another way of thinking about these effects. Trailer style booms and whooshes are some typical examples. Any sound effects that don't fit into the reality of the scene, but rather are used for emotional impact, would fit into this category.

## Final Mix

The final mix is where all the ingredients (dialog, voice over, music, and sound effects) are put together into a polished soundtrack. This is a complicated and delicate process, but some basic elements are as follows:

- Balancing the different elements of the soundtrack with each other

(both through volume and the high/low frequency content of the elements)

- Establishing dynamic peaks and valleys in the mix (this can add excitement and reinforce a project's pacing)
- Spreading elements throughout the stereo (or surround) field through panning (in the case of post production audio, this can also mean tracking objects on screen)

## Mixing Formats

There are various formats a final mix can take, depending on the ultimate delivery needs. Stereo and 5.1 are some of the most common today but, VR (Virtual Reality) and AR (Augmented Reality) formats are on the rise.

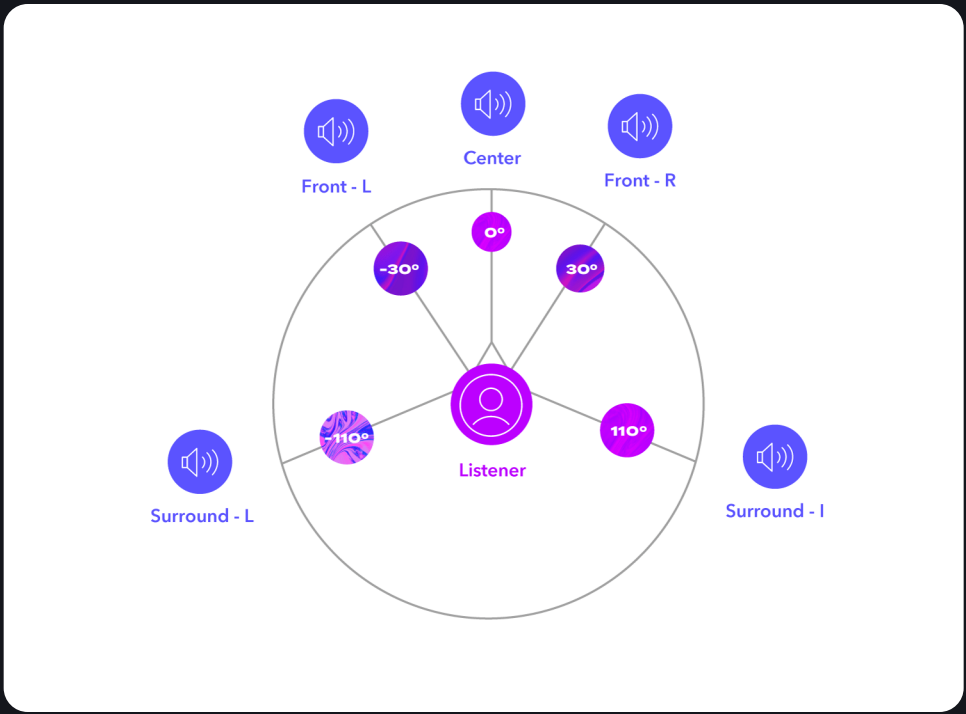
Below is a quick breakdown of some key differences between popular formats:

Mono – (a single audio channel) This format is mainly used for stems like voice over or dialog. Some public networks still require a mono mix for broadcast, and it is a good idea to make sure your mix is mono-compatible for playback on phones, smart home speakers, and other mobile devices.

Stereo Mix – (2 audio channels) *A left and a right speaker*. The most common format outside of a theater environment. The vast majority of web and broadcast content is in the stereo format. All mixing facilities are set up to work in stereo.

5.1 Mix – (6 audio channels) *Left, Center, Right, Left Surround (Ls), Right Surround (Rs), and Low Frequency Effects (LFE)*. The "5" in 5.1 represents the Left, Center, Right, Ls, and Rs channels. The ".1"

represents the discrete low frequency effects channel. The 5.1 format is commonly used at film festivals, in broadcast television, and for many online streaming services. Most post audio mixing facilities are set up to work in 5.1. This format has been around for some time, but is highly immersive and remains one of the most popular surround formats.




An example layout of a 5.1 audio mix.

7.1 Mix – (8 audio channels) *Left, Center, Right, Left Side (Ls), Right Side (Rs), Left Rear (Lr), Right Rear (Rr), and Low Frequency Effects (LFE)*. This format is used in most modern cinemas. Many, but not all, mixing facilities are set up to work in 7.1. This format shares many characteristics with the 5.1 format, but 7.1 provides more separation in the surround speakers around and behind the audience.

VR – (played back on headphones) This is a new and developing field. Essentially, a virtual reality mix represents a full sphere of sound. VR video players have special decoders that tailor the audio experience depending on where the viewer is looking. For example, if a dog is barking to the right of the viewer's perspective, the sound will come from the right side of the sound field. If the viewer turns to

face the dog, the sound will come from the center of the sound field. It is similar to the way audio works in video games.

There are many ways to create virtual reality content. Some methods use a spherical video along with a 4 channel audio track, and some methods use special programming and are developed inside a game engine. Major VR platforms include Facebook, Youtube, Oculus, and Vive.

Dolby Atmos / DTS:X -These are next generation spatial mixing formats where sounds are treated as distinct moving objects throughout a 3D space, rather than just panning between preset speakers. This allows for more consistent playback on different speaker arrays from 2 to 64 channels. Currently, these formats are most commonly used in big-budget, wide release films, and require a specialized cinema (or home theater) setup to properly playback. Mixing facilities require specialised systems to work in Dolby Atmos, so be sure to confirm a facility's capabilities if this format is needed for your project.

Which format should I choose? – Not all projects require the maximum possible number of speakers. Stereo is generally the default format. Many films also sound great in stereo, especially if they are primarily intended for internet playback. If you plan to present in a theater (like at a film festival or via major streaming service) expanding to 5.1 may help. Talk to your audio facility about your needs.

## Review and Revisions

Many studios offer in-house reviews, but online reviewing methods are increasingly being used. Frame.io is a great online tool for reviewing an audio mix remotely. It allows the filmmaker to drop in notes at specific points in the timeline, streamlining communication with the mixing house.

## Deliverables

Depending on the nature of the project, a variety of deliverables will be needed from the post audio team. The most important deliverable is, of course, the finished mix (possibly in multiple formats, as needed for the project). Stems/splits are also commonly required. These are isolated elements of the mix (dialogue, voice over, music, sound effects) that can be helpful for reversioning.

It is important for the filmmaker to communicate any unique needs. Post audio can streamline their workflow for each project if they know exactly what deliverables/formats are required before work begins.

## Audio Specifications (aka Specs)

Specs are guidelines to ensure consistency in loudness across multiple programs, as well as to prevent undesired audio issues (clipping, digital pops, etc). This idea applies to broadcast television

as well as to online streaming, although the technical specs are different for each one.

It is important to understand what environment your project will be playing in and what specs that environment may require. If ignored, a project may fail quality control in preparation for broadcast and be unable to air until addressed. This scenario is primarily applicable to broadcast television, where government regulations specify acceptable loudness levels (these specifications vary from country to country). Individual networks may also have unique spec requirements that post audio should be made aware of.

Even outside regulated environments like broadcast TV, there are spec recommendations for a variety of listening environments that can help ensure comfortable and consistent volume levels for listeners.

## Measuring Loudness

There are numerous loudness tools/plugins available that measure all aspects of a completed audio mix. The two most scrutinized measurements of any audio mix are the average loudness for the entire program, and the level the loudest sound.

Quick definitions for common measurements/tools used:

LUFS/LKFS – This is a standardized unit of measurement of perceived loudness. It is used for finding the average loudness of an entire audio mix; this is referred to as Integrated LUFS/LKFS. LUFS and LKFS used to operate with slight differences, but today they are used synonymously.

True peak – The measurement of the loudest moment of the mix.

True peak Limiter – A specialized tool/plugin that allows the user to set a loudness ceiling that the audio cannot exceed. True Peak Limiters are sometimes confused with "Brick Wall" limiters. In short, True Peak Limiters are much more precise.

## Loudness Recommendations

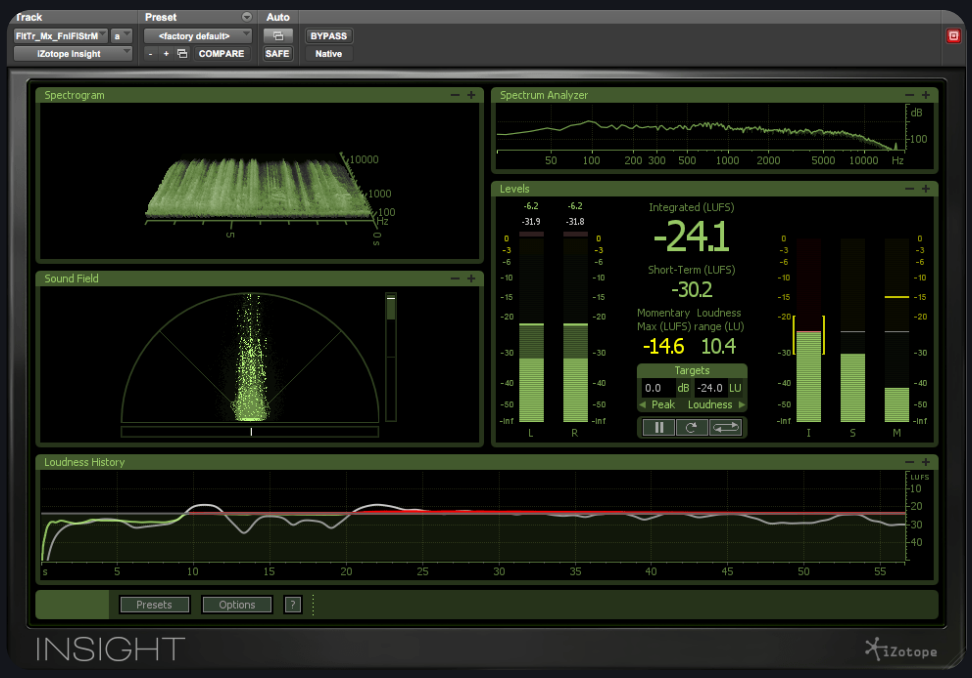
Below are some general recommendations for a variety of common listening environments. As mentioned earlier, if your project is playing in an environment that may have unique and strict loudness guidelines, make sure you are aware of them and communicate this with your post audio team.

Web – A common target for web mixes are Integrated LUFS/LKFS of -16db (+1/-1db) and True peak of -1db. There are no strict regulations on web mixes, but these recommendations should yield a product that is consistent with other web content, while still leaving room for a dynamic and engaging mix.

Broadcast – Television is arguably the strictest environment regarding loudness. In the US, typical measurements are -24db (+1/-1db) Integrated LUFS/LKFS and -10db True Peak (some networks provide a bit more wiggle room, but these recommendations should cover all the bases). Different countries have different regulations, so be sure to check what measurements are appropriate for your project.

Some networks also require specific splits or versions of the final mix. You can usually find a network's audio technical specifications online, or by requesting that information from the network directly.

Film – There is no standard or regulated loudness standards for films. Films are ideally mixed in precisely calibrated rooms that allow for loudness to be mixed by ear. In other words, you are free to use the full dynamic range that digital audio can offer, as long as the film is mixed by an engineer with an accurate frame of reference. It is also wise to keep in mind the venue your film will be played in. For instance, many film festivals have a surround system. In these cases, it's important to know what format that festival requires/accepts (Stereo, 5.1, 7.1, etc).




One final note. It is common to request multiple final mixes on films. If your film is to be shown at a film festival with 5.1 surround sound as well as online, you may want to request a separate mix for each. In addition, while film festivals may offer 5.1 surround, they rarely *require* it. Your film may work better in a stereo format. It all depends on the sonic needs of your project (for wide theatrical releases however, a surround format is usually universally required).

In short, different environments have different optimal formats and

loudness settings. Communicating your needs will help post audio provide the best possible mixes.

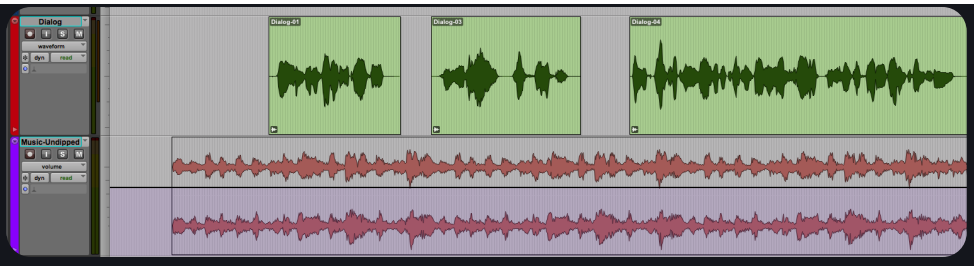
VR – There are currently no standards for loudness in VR, but the Web loudness levels mentioned above are a decent starting point. There is a variety of virtual reality formats, each with its own unique spec requirements. It is important to read up on these for whichever format you are using. Youtube VR and Facebook 360 are some of the most popular VR formats (spec requirements for both of these can be found online).

## Stems/Splits and how to use them

Stems and Splits are interchangeable terminology. They both refer to individual elements of a final mix that have been *split* out into their own files (voice over, dialogue, music, sound effects, etc). Stems/splits are helpful for processes like language versioning or future repurposing of content. For example, by using an MDE split (explained below), editors can easily swap out an English voice over for a Spanish voice over without recreating the whole mix. It is important to specify exactly what stems/splits are needed with post audio so they can best provide the needed deliverables.

The following is a list of common deliverables to request from post audio:

1. Dipped Music – Contains any volume automation added to it during the mix. Dipped music represents the music exactly as it is in the full mix.
2. Undipped Music – Does not contain any volume automation added during the mix. This can be helpful if you need to remix the project later due to major changes, or if the project needs to be repurposed elsewhere or into a language that requires more or less time to say the same thing.

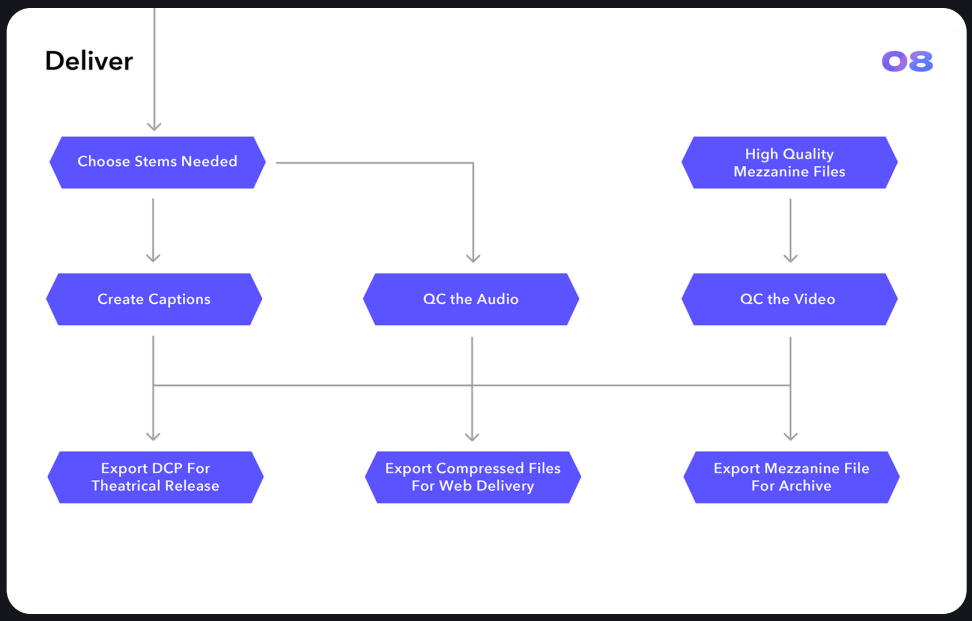



An undipped music stem like this allows for international versions of the edit to be adjusted based on the difference in dialog length for any given language.

CHAPTER 08

# Delivery

At the final step in the process, each piece of media is packaged together and optimized for each medium: web, broadcast, theatrical, mobile…




This section was contributed by Michael Kammes, Creative Technologist.

File based media delivery has become the industry standard over the past ten years. Tape deliveries have become antiquated and typically only requested for redundancy or for use by legacy systems.

Often, you'll find file delivery specifications in a "deliverables" document. Strict adherence to the deliverables document is essential to ensuring your media is accurately played back and viewed in the way in which it was intended. Failure to adhere to the specifications defined within a deliverables document may lead to picture artifacts, sound hiccups, or failure to play all together.

Advanced delivery specifications may also include incorporating strict naming conventions of the media files, specific audio channel mapping, the embedding of metadata, and wrapping of various media files into "wrappers". Lastly, the way that you transfer files from one location to another can take the form of traditional FTP, web-based transfer services, P2P (peer-to-peer) applications, or

dedicated accelerated transfer utilities.

## End point determines the format

Each video endpoint requires your media to be encoded in such a way that their data center can process the media effectively. As a general rule, the more consumer in nature the end point is, the more flexibility there is in the deliverables.

As a reminder, all professional outlets will have a deliverables document or webpage to reference – ALWAYS ask for this prior to creating deliverables.

Social Media

Social media websites, such as YouTube, Vimeo, and Facebook, can handle SD and HD resolutions, and in some cases, 2K, 4K and above. Each platform may differ in aspect ratio, and typically prefer compressed media, such as those media files generated by the cameras found in many mobile devices. These sites encourage smaller file sizes for faster uploading. However, many of these sites will accept a broad range of less compressed formats, albeit not advertised. It is also not uncommon for social media sites to perform a subsequent re-transcode of your uploaded file to ensure it will play back from their servers appropriately.

Broadcast

Broadcast outlets, such as NBC, CBS, ABC, FOX, PBS, and others have strict guidelines for delivery. This is mainly due to the fact that these are the media files that the Network Operations Center

(NOC) will play when on-air (they won't be re-compressing it first). As the media playback systems for broadcast centers are purpose built, the flexibility for variations in media formats is drastically reduced. In addition, broadcast outlets routinely run submitted media files through Quality Control (QC) applications to verify conformity to the defined specifications.

Feature Films (mainly Netflix)

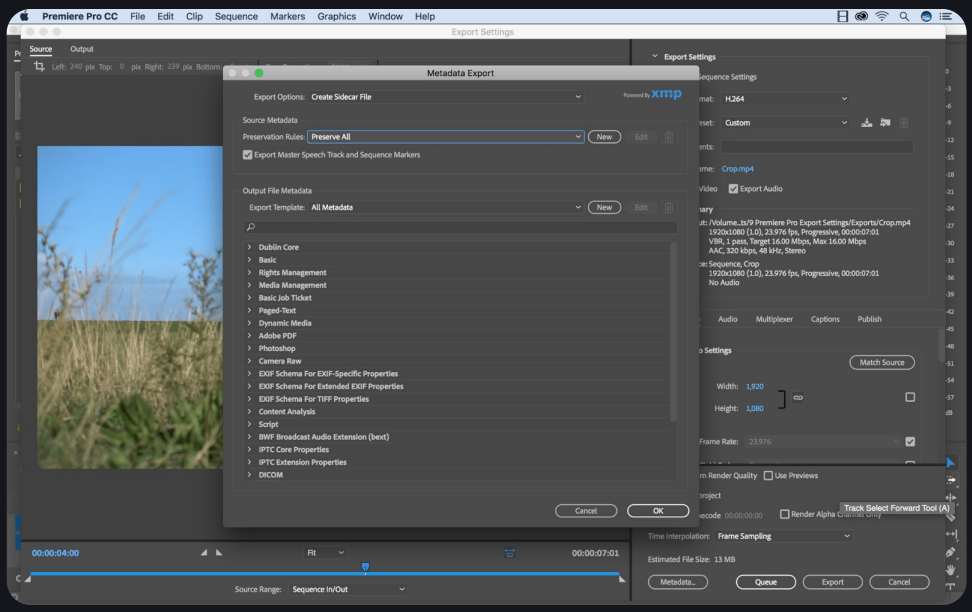
Non-broadcast professional video outlets, such as Netflix and Hulu, have defined their own standards for media delivery. As they are not beholden to traditional broadcast infrastructure and legacy set-top boxes, they can request higher fidelity media. This media is often very large, as VOD outlets will create compressed streaming versions for consumption by viewers on a wide variety of devices. The increased quality of these submitted files also allow for future manipulation and usage as newer formats gain consumer adoption (HDR, UHD/4K, etc.).

## Creating the files

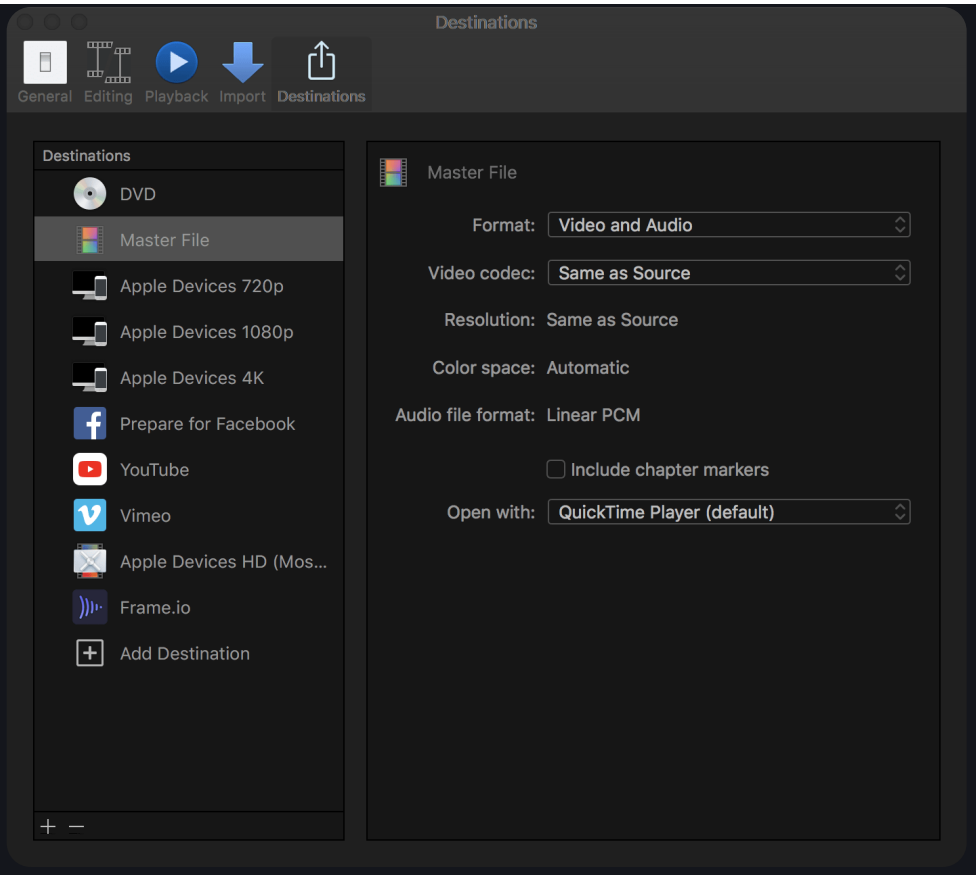
### What type of video?

The key to a good looking end deliverable is to create your final deliverable(s) from the highest quality master file. Re-compressing media files that have already had a round of compression will yield a worse looking media file. Much like making a photocopy of a photocopy, generational loss can occur with each subsequent re-compression, so it is important to start with the highest quality file available. The highest quality file, from which all other versions will be generated from, is often called a "mezzanine" file.

The easiest way to create a mezzanine file is to export from the NLE project itself. This means either exporting from within the NLE or, exporting through a software application that understands a project file, such as Adobe Media Encoder with Adobe Premiere projects. Accessing the original media when creating a deliverable means you are generating a mezzanine file from the highest quality files available – the source footage.




Adobe Premiere Pro Media Encoder




## What type of video codec should my mezzanine file be?

A good rule of thumb is to create a mezzanine file in the highest quality you can without exceeding the quality of the highest resolution of video in your timeline. For example, if you shot something on your cell phone (H.264), exporting in an uncompressed format would be a waste of drive space. You can't add image fidelity to media where it wasn't originally present. Imagine taking water from a small glass and pouring it into a larger glass. It's the same amount of water – just in a smaller container.

## What if my source media is very compressed?

The exception to this is if all of your media is compressed – perhaps

all shot on a cell phone (H.264) – then exporting to a cell phone format would further reduce the quality of your file. This is because lightweight codecs like H.264 are Long GOP and extremely lossy, and thus tend to breakdown following subsequent rounds of re-encoding. In this case, we will want to create a non Long GOP mezzanine file, that is as perceptually lossless as possible. Often, this becomes a mezzanine file in a ProRes or DNxHD/DNxHR codec. On the higher end, formats like image sequences (DPX, TIFF, J2K, etc.) are another popular choice.

As the industry moves to UHD/4K and HDR, new formats and codecs have been developed to handle the increase in image fidelity for video endpoints. While ProRes, DNxHR, and image sequences are well equipped to handle this, what formats do we create for delivery?

Delivery Codecs

When streaming, HEVC and AV1 are the newer competing compression standards for UHD/4K video. HEVC, also known as h.265, is the "next generation" of H.264/AVC encoding, and promises to cut file size by 50% or more compared to H.264 encoding. Unfortunately, there is licensing involved with h.265 usage, making wide adoption to be much slower than anticipated. AV1 was born from this licensing debacle, and companies like Amazon, Google, and Netflix have moved towards this royalty free codec. AV1 also promises an additional 30% file savings compared to HEVC. As the "4K formats war" continues, it's important to always check with your VOD partner o what standard they are utilizing.

For UHD/4K broadcast outlets, XAVC appears to be the new "standard". As there is no formal broadcast codec for UHD/4K formats, XAVC has been engineered to be a trade-off between quality and ease of playback. This is accomplished via "classes" (of quality: 50, 100, 200, 300, 480) . XAVC is

actually the highest quality level (5.2) of the H.264 standard you're accustomed to. This means it's a bit easier for current NLEs (and the computers they run on) to work with. Many cameras will record into this format, too. XAVC for broadcast is much like how XDCAM was for broadcast in 2005. These files are traditionally in an OP1a MXF wrapper.

Broadcast outlets may also request 720p or 1080i DNxHD/DNxHR, or ProRes, or XDCAM in an OP1a MXF wrapper as opposed to the newer XAVC format. The changeover from traditional HD broadcast outlets to ATSC 3.0 ("next gen" T.V.) is just beginning, and not all facilities may be able to accept UHD/4K material.

Of course, you should always make sure your deliver codec meets your desired level of image quality. Check out the Frame.io Codec Comparison to see how different codecs affect your footage.

A vast majority of video endpoints and broadcast NOC outlets look for video in a traditional 16×9 aspect ratio. However, the rise of video acquisition with mobile devices, plus consumption of those same videos on social media sites like Instagram have led to the need to deliver media in different aspect ratios. This can include: Landscape (1.91:1), Square (1:1), and Vertical (4:5).

## What type of audio?

Video without audio is just surveillance! That's the last thing we want. So, we must encode our audio in a way that not only retains the audio fidelity of your project, but also so it can be decoded and played back by the viewer.

Perceived audio quality, just like video quality, can vary incredibly on where and how the end user decides to view the media. Is it being heard through a small mobile device with headphones, or in a movie theater? Each media outlet will manipulate your audio. This manipulation can be in the form of file size compression (bitrate) or tonal compression (dynamics). Both can affect the artististry in your project, so it's important to create your files audio tracks properly.

## Audio layout in your NLE

Most NLEs support mono and stereo audio editing. Some even support various versions of editing or mixing surround (5.1, 7.1, etc.). However, most NLEs understand that professional audio

manipulation isn't best suited for an NLE toolset. The video editing toolset is prominently frame based and centered around video augmentation, not audio.

Because of this, it's common practice to migrate the audio work to an audio centric NLE, also called a DAW (Digital Audio Workstation). Often this migration is in the form of an AAF file (not to be confused with the audio format AIFF). Audio is then worked on within the DAW, and then imported back into the NLE. The audio and video files are then married/exported into a mezzanine file, as noted above.

Like video mezzanine files, audio mezzanine files shouldn't be at a higher quality than the source media. However, if the source audio media is highly compressed, then exporting into a perceptually lossless format is preferred. Also, because audio takes up far less storage space than its video counterpart, it's not uncommon to convert audio to 16-bit or 24-bit WAV or AIFF files, at 48kHz (or better).

Audio Placement

"Placement" of your audio can take many forms. This includes what audio segments get played through what channel (left/right, front/rear, etc.), as well as placement in the file (how the audio is encoded with the video).

For audio placement within the correct channels, it's important to follow the placement standards. A common layout is the SMPTE surround standard: L R C LFE Ls Rs (Left, Right, Center, Low Frequency Effects, Left Surround, Right Surround.) However, this may vary depending on your deliverable requirement.

Many media outlets also require a stereo mix, a surround mix, as well as foreign language tracks.This means you could have

close to ten or more audio tracks for each video file!

Audio channels, as described above, correspond to the artistic organization of the audio sources, e.g. what speaker the audio is heard from. Audio streams on the other hand, are the places within the digital media file that the audio is encoded. For example, one audio stream in a media file can contain 6 surround channels of audio.

Often, the timeline in your NLE won't be laid out in the exact way that your end deliverable is being requested. That's OK! This means you will have to map your audio to the appropriate channels as outlined by the deliverables doc that you've requested. Most NLE's have a way to map audio tracks (or roles) to the appropriate output channel and stream.

## There are audio laws. Yes, legal ones.

I'm sure you've watched a show on T.V. or online, followed by a commercial or subsequent video, and found that the volume between the two are completely different. This is a major problem, not only for the enjoyment of the media you're consuming, but because hearing damage can potentially occur at high volumes. Due to this, the CALM Act (Commercial Advertisement Loudness Mitigation) was put into law by the FCC to ensure that broadcast television aired content at similar volume levels. Mixing to be in compliance with the CALM Act is important to ensure your audio isn't rejected by the NOC.

While there is no current enforceable CALM equivalent for online-only video, it's good practice to adhere to the same standards. This allows viewers who are consuming both (re) broadcast media from television, as well as exclusively online content to enjoy the content without constantly adjusting the volume during playback. It should

be noted, however, that may online platforms attempt to do this for you, but without your knowledge.

Loudness Settings for Youtube

YouTube will adjust the playback level of your audio. YouTube calls this "Loudness Equalization". You can see this by right clicking on the YouTube video playback window, and click "Stats for Nerds". This is what the "Volume/Normalized" heading is. YouTube analyses the audio present in your video and makes a determination if the volume at certain places – or overall – in your video needs to be adjusted for a pleasurable volume level of playback for the viewer.

Unfortunately, there is nothing you can do to prevent this. However, what you can do is minimize the audio tweaking YouTube does so as much of your artistic decision on loudness is retained.

You can start by metering and mixing your audio using the Loudness Units relative to Full Scale, otherwise known as LUFS, inside your NLE or DAW (you may need a 3rd party plugin to monitor audio in LUFS.) YouTube doesn't release exact values, but YouTube tends to shoot for -12 to -14 LUFS, so try and keep your audio mixed within that range to reduce the amount of tinkering that YouTube does.

That being said, aiming for a specific integrated loudness doesn't work, reliably. Experiment and see what sounds good to your ears.

## Closed Captioning

Captioning, both open and closed, serves many purposes. Open captioning, or timed segmented transcriptions and descriptions of the audio, are permanently "burned" into the video frames of your finished media and are thus always visible. Closed captioning (aka "CC"), on the other hand, follow the same basic concept as open captions, but are optional viewing for the end user. In essence, these closed caption files can be toggled on and off by the end user if the media player supports it. The key to giving a user this functionality is to not only ensure the media player supports closed captioned media files, but also embedding (muxing) the caption files into the correct place within the media file and its wrapper so it can be accessed.

## What type of captioning?

The most common use of closed caption files is for various languages. This allows your media to be consumed and comprehended by viewers who don't speak the language spoken in your video. Many VOD outlets will request one or multiple translations to broaden your media's reach.

Standalone caption files can take many formats. These formats have varying degrees of usefulness: some handle visual formatting for the text, with options such as bold, underlined, or *italicized* text. Many

flavors even have various fonts that can be used, plus placement of the text on the screen. While there are dozens of formats in use, let's cover the most frequently used formats for web and broadcast.

- SRT (SubRip)
- VTT (WebVTT)
- SCC (Scenarist Closed Caption)
- TTML (Timed-Text Markup Language)
- STL (EBU-STL)
- TDS (Caption Center)
- RT (Real Text)
- SMI or SAMI (Synchronized Accessible Media Interchange)
- CIN (Captions, Inc.)
- ASC (Cheetah ASCII)
- CAP (Cheetah Binary)
- CAP (NCI Binary)

I know! There are a ton of formats! Luckily, most closed captions file are nothing but specially formatted text, so a quick internet search will find many free sites that can convert between the formats. If your endpoint is requesting captions, they will outline the format they require.

Once you have your files captioned in your NLE, how to we pair the captions with the media for your deliverables? This is where you need to make the decision – will your caption files be embedded, or sidecar?

## Embedded or Sidecar?

### Embedded

While an all-in-one embedded file sounds like the most complete option, it's not always the easiest. This is because of the wrappers the media resides in. A Quicktime MOV wrapper, or an MXF wrapped file (for example) have different specifications as to where the captioning is referenced within the wrapper. This can also differ if the media is standard definition or high definition. If the end point or VOD outlet doesn't support that wrapper, or the caption file is not embedded correctly, the closed captioning files may not be found by the media player. An embedded closed caption file also reduces the flexibility you have with that file, as it's formatted in only one way – that is, how the media file wrapper requires it.

### Sidecar

A sidecar captions file is a standalone, self-contained file that is meant to be referenced by a media player that understands the relationship between the closed caption file and media file. This is primarily done through the timecode present in the media file, and its matching timecode within the caption file. The media player will access both and sync the playback during viewing.

## Considerations

While providing a captions file for optional viewing seems to be pretty self explanatory, there are some considerations to be made.

If your media is going to social media outlets like Facebook, many users have the audio disabled for autoplay videos. This means as users scroll down their timeline, they will not hear your audio by default. Many video producers will burn open captions into their video, so the message behind the video is conveyed without

audio…or to entice the viewer to enable the audio.

YouTube, and its parent Google, suggest that media files which are captioned may improve their search engine ranking or even strengthen their applicability to various searches. In short, captioned videos may receive an SEO boost. The exact amount of usefulness provided by captioned files is not public information, however.

YouTube allows for users to upload caption files, too. Plus, YouTube will try and create its own caption files for uploaded videos. Often, this type of automated captioning is less than optimal. Poor audio quality, accents, and background noise can interfere with the YouTube algorithm. So, it's best practice to caption your files manually or use one of several online providers who will transcribe your video for a cost per minute.

## Check for Quality Control

Quality control, also known as "QC", is the method of checking your deliverables to ensure they meet the requirements outlined by the deliverables document from your endpoint. QC is not only a visual "check", but also includes scanning the files for technical issues like legal audio and video levels, proper timecode, and file metadata, just to name a few. For broadcast, QC is largely an automated process, where specialized software applications QC your media, and generate a report with issues and their respective timecode values. Note that visual QC is subjective as opposed to a "pass/fail" stamp.

Failing QC can be a time consuming and expensive proposition. Most media created for broadcast are on tight production schedules. If a file fails QC, this means more time and money is spent to remedy the failure. It's also quite embarrassing! This is why it is so important to review the media closely before submitting the

files to the endpoint.

Quality Control can encompass hundreds of potential issues. Some common QC flags include:

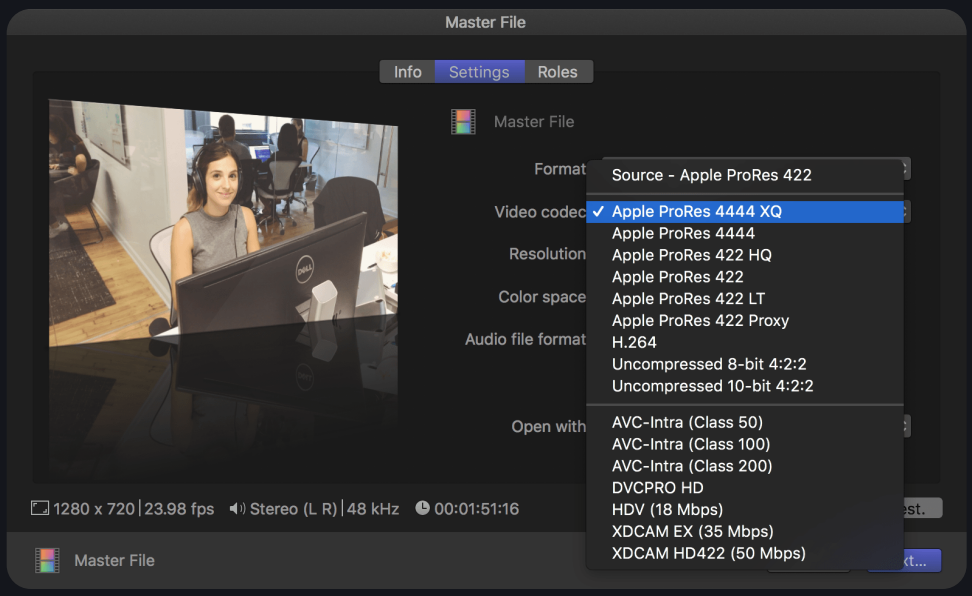
- Bad/dead pixels
- Visible production equipment
- Compression Artifacts
- Compositing Errors
- Luminance Shift
- Color levels (examples: Rec. 709 vs. Rec. 2020)
- Incorrect audio levels and noise
- ADR (levels and sync)
- Incorrect timecode and frame rates
- Missing caption files
- Incorrect wrapper, audio placement and language tracks
- Poor quality footage (stock or low res source material)

In addition to the visual QC process, many video endpoints have automated QC applications to check the media files for potential issues that may not be seen visually. Baton and VidChecker are common QC tools for this purpose. In response to this, many media creation facilities have adopted the same QC tools for use in-house to avoid the failing at an outside facility. Many of these tools also have the ability to “fix” the media file errors they find. However, these automated QC systems may not recognize that when a flag is triggered, it is an artistic decision as opposed to an error. This is another reason why a visual QC is also strongly encouraged.

# Wrap it up and export

You have your picture locked, and your audio is finished and married to your video in your project. Now: how do you get your finished project out to the world? There are a few ways to accomplish this.

First, as we discussed earlier, exporting a mezzanine file from your NLE or finishing system is preferred, so you have a master file with which to create your various deliverables versions from. This ensures that you have the highest quality to start from. The next step is creating the individual versions for your delivery. We now have several options and tools available to create these files.




## Hardware or Software?

Transcoding can be done either “in software” on a Mac, Windows, or Linux operating system with a specialized transcoding application, or it can be done “in hardware”, which requires the user to play the video out of the computer via “baseband” connections

(video cables).

### Hardware Transcoding

This method requires the usage of baseband connections, that is, SDI cables, or even legacy component or composite connections. Hardware transcoding also requires a video I/O device, such as those offered by AJA Video Systems or Blackmagic Design. The cables are connected from the video I/O output of your editing/finishing workstation into a purpose-built transcoding "appliance" which will create your files.

While this process is real-time, encoding appliances are traditionally more expensive than similar software versions.

As a general rule, hardware encodes tend to look slightly better than software transcodes, but hardware transcoders tend to lack the flexibility of software transcoders

### Software Transcoding

Software transcoding is by far the most accessible solution for must users because it can run on whatever computer you already have. However, it is not without it's "gotchas".

Codecs, like music, can be licensed. If the codec is not freely useable by the public, then the manufacturer needs to license the codec for use by a software manufacturer. Such is the case with Apple's ProRes codec. While the ProRes codec can be encoded and decoded on virtually any Apple computer, Apple has not issued a blanket license to Microsoft Windows. This means video software manufacturers need to request a

license from Apple. Unfortunately, many software transcoders have incorporated unlicensed Apple ProRes encoders within their software. Unlicensed ProRes implementations have no quality control and may fail QC. For professional deliverables it is strongly recommended to use licensed ProRes implementations.

Many software transcoders will leverage your GPU, or Graphics Processing Unit (your graphics card) to speed up video transcoding. While the speed increase is a plus, performance (and quality!) can vary wildly between graphics cards and transcoding softwares. Always check to see if your transcoder will leverage your GPU.

Robust transcoders may also provide for distributed transcoding, that is, using network resources to split up your transcode task and leverage the speed of multiple computers connected to your network. These systems usually require a fair amount of setup to optimize, and often fail to transcode as quickly as anticipated. The process of splitting up a transcode into many smaller pieces, then recombining those pieces often yields diminishing returns. Distributed transcoding in bulk is traditionally more useful than accelerating a single file transcode.

Most basic transcoding tools require three things: a source file, the parameters for the transcode, and a destination location for the finished file. Some basic transcoding tools include: Apple's Compressor, Adobe Media Encoder, and Handbrake.

Larger scale or enterprise transcoders have additional features which are useful when large amounts of transcodes are needed. These features may include: public and private watch folders, decision-based transcoding that is dependent on the audio, video, and metadata present in the file, embedding/de-embedding of closed caption files, automated QC, and our next topic: actual delivery of the files.

Common tools include Telestream Vantage and Root6 Content Agent.

## Send Off The Files

Now that your media files are created, we need to deliver the files to the end points. Most VOD platforms have web page based upload forms. For groups of files, or large files, however, web-based (HTTP) uploads can be slower and less reliable than optimized connections. HTTP is also not as secure as other transmission methods, meaning your media stream could be intercepted. To combat this, we have purpose-built media transfer tools.

FTP, or File Transfer Protocol, is a legacy way of transferring files that offers a bit more resiliency than traditional HTTP methods. However, FTP is a two step process – upload to a server, and download from that server to another system. This means that transfer time may be longer than a direct peer-to-peer transfer.

Dedicated file transfer services, like Aspera, Signiant, File Catalyst, and offer accelerated and secure file transfers between locations. These transfers are routinely faster than FTP or HTTP, and some even meet MPAA (Motion Picture Association of America) certification or the newer TPN for security. These transmission methods are considered facility-based tools, and thus are more expensive than FTP or HTTP methods.

Frame.io also offers accelerated file transfers alongside its collaboration toolset. Although Frame.io runs on standard HTTPS, its speeds match or exceed those of the dedicated file transfer solutions, and it also offers enterprise security as audited by TPN.

CHAPTER 09

# Case Studies

We walk you through the details of a range of different workflows, from Hollywood tentpoles to wedding videos, so you can see the tradeoffs.

*1 minute read*

Thank you to all our case study contributors, and to our writer, Lisa McNamara!

Now that we've covered all of the main concepts and processes of post-production, it's time to get some real-world experience. As you've learned, there is no single best workflow. Each project's unique circumstances requires a unique workflow design.

So we sat down with the creative and technical teams on a series of popular projects from different sides of the professional video industry, to learn the details of their workflow. The budgets and timeframes vary widely, from a months-long Hollywood tentpole to a weekly YouTube show, but each of the teams we spoke with is at the top of their game, doing killer work.

Blockbuster – Mission: Impossible

Mission: Impossible – Fallout isn't "just" an international blockbuster—it's also one of the more complex post-production workflows we've seen. Shooting on film and digital, and editing in 4K with IMAX and stereo deliveries, it doesn't get much more intense than that.

It's no surprise, then, that the large editorial crew relied upon a supremely organized and highly technical workflow to get them through such an ambitious project. But even if you're not shooting a multi-million dollar movie, there are takeaways contained herein that can inspire your approach to any project.

## The overview

Editor Eddie Hamilton A.C.E. and his core editorial crew of two first assistants, two VFX editors, two second assistants, a stereo editor, and a trainee (plus another twenty-plus post-production crew members) brought Mission: Impossible – Fallout to the big screen despite having to shut down production for two months after Tom Cruise's stunt accident—and not getting an extension on delivery. Add to that an action sequence that required taking 70 hours of footage down to 7.5 minutes, locations from New Zealand to Paris to Abu Dhabi, and a test screening scheduled for four days after shooting wrapped, the post-production process itself was its own sort of impossible mission.

## The breakdown

Director Chris McQuarrie and DP Rob Hardy elected to shoot mostly on film, using Arriflex 235 and Panavision Panaflex Millennium XL2 cameras. The format was 35mm anamorphic, at true 24fps (not 23.976). Additionally, for the helicopter

sequence (the aforementioned 70 hours culled to 7.5 minutes) they used the Panavision DXL 8K digital camera because they could run for 40 minutes without reloading– important when you've got multiple helicopters in the air. And for the HALO skydive sequence they used a Red Weapon 6K camera (strapped to the camera operator's head!).

While many large-scale features are still edited at 1080p, the Mission crew worked at 4K throughout. The rationale was that 4K DNxHR LB files, at 144Mbps, were only 25 percent larger in file size than the DNxHD 115 1080p codec they had used previously, but with four times the resolution. With a visual effects load of some 1600 shots, the additional resolution made that part of the workflow easier, as well.

Film was processed at Cinelab (London) and scanned at Pinewood Post on a Scanity at 4K for an eventual 4K delivery.

While it's common for the data to be held at the DI (digital intermediate) facility (in this case Molinare in London) the volume of material for Mission was so large– and Molinare had other large-scale projects working concurrently–that another storage solution became necessary. Fluent Image (also in London), therefore, was responsible for holding all data and distributing it to Molinare and to Double Negative, the main visual effects vendor.

## Testing

If you ask any of the crew what the most important part of the workflow was, they'll tell you that testing was the essential first step. Because the work was divided among numerous vendors, everything had to be tested thoroughly to ensure that everyone was exchanging files in the correct formats and color space–and that the technical and communication channels were working without any hitches.

## The hardware

Eddie had two different primary computing setups: a dual-screen desktop that he used in the main edit suite and a laptop for when he was on location or traveling. There was also a third setup installed at Chris McQuarrie's house so that he had full access to the work-in-progress when not on the set or in the cutting room.

The stationary edit suite was equipped with a 2160p 65″ Panasonic OLED TV connected by HDMI to an Avid DNxIO box attached to a 2013, 6,1 Mac Pro. They had a 48 TB Avid ISIS for storage and Eddie used a Razer Naga gaming mouse, which he customizes frequently according to the tasks he's handling. For audio, he de-embedded the HD-SDI audio and fed it to a digital 5.1 speaker system using Dynaudio Air 12s.

The mobile suite ran Avid on a 15″ MacBook Pro with the Duet App active so that an iPad could be used as a second screen, and the Razer mouse, iLoud Micro Monitors, a 32 TB G-speed Q Raid and two 4 TB LaCie Rugged Raids were hooked up to a USB 3.0 hub. Eddie also used his laptop with an encrypted SANDISK Extreme Pro 512 GB SD card to watch dailies and make selects while he was traveling, whether by air or by car, to utilize every waking hour–both because the workload was so immense and because he likes to watch absolutely every frame of footage in his search for hidden nuggets.

## The dailies workflow

Film was developed each night at Cinelab and delivered to Pinewood Post early the following morning. Pinewood

commenced scanning immediately, with the scanned DPX files given a basic color correction and then transcoded to 1080p DNxHD36 MXF files, so they were small in size and could be quickly uploaded to the cutting room for an initial review.

Some of the metadata came straight from the lab, but early each morning the trainee manually inputted additional information including focal length, selects, and a description of the action. The first assistant distributed sub-clips to the second assistants on either a camera roll or scene-by-scene basis (depending on the order of the clips coming from Pinewood), and they created the Avid scene bins. Meanwhile, the first assistant generated a single sequence of all the dailies for Eddie so that when he got in he could immediately see what had been shot the previous day in case he needed to address any questions from the set.

While the early morning crew was handling that initial upload, Pinewood's dailies colorist would do a full color pass and new DNxHR LB 2160p (4K) MXF files were shuttled to the cutting room on a drive. In the background, one of the assistants would copy over the DNxHR files onto the ISIS and then, at a convenient moment for Eddie, all scene bins would be shut, the DNxHD36 files archived, and the master clips would be relinked to the new 2160p media and the relevant sequences refreshed. When Eddie reopened the bins he'd been working in, he'd have the fully graded UHD dailes.

The assistants were responsible for tracking, logging, and exporting the clips and metadata to the codebook (a Filemaker database created by the VFX team that all post-production team members could access to track any element or process throughout the production), and for the studio and production dailies.

Pinewood also supplied the CDL (color decision list) metadata for each clip, which was also visible in the Avid bins

and the code book.

The workflow varied when Eddie was on location. For example, in New Zealand (where they shot the helicopter sequence), one of his assistants traveled with him, and Pinewood put a mobile setup next door to where Eddie was working. While there, he initially worked with the footage that came off the video tap of the film camera (while the film was flown to and processed in London), while Pinewood transcoded the digital IMAX files on site. Once his assistant received the 4K files from London, he would have to eye-match to replace the video tap files with the proper 4K media.

## Organizing the material

Eddie is a firm believer in being able to try alternate versions rapidly, so naming conventions and timeline organization are deeply detailed.

They used a rather straightforward numerical folder structure for the various tasks (cutting copy, scene bins, VFX, music, sound effects, etc.), and in the scene bins folder were subfolders for the scenes from 1-20, 21-40, and so on. The slate letters from the take were added to the scene numbers in the bin name, along with a brief description of the scene.

The assistants broke down all the dialogue scenes in a very detailed way, into massive line strings where all the line readings for each line in the script are broken down by wides and mediums and overs and tights. That way, when the director asks Eddie to see the options for the lines, he can skim through, show him the shot sizes, and audition the various readings quickly, which equals fast progress. This was especially handy when working on the set, as the director could spend an hour here or there during lighting setups or meal breaks to work with Eddie.

In terms of the audio workflow, for each take there were multiple BWAV tracks with a channel 1 mix track. Eddie works with all the tracks in the timeline, and the assistants solo channel 1 so when he presses play he's only listening to the mix track, but all the other tracks are there in the subclip. When he assembles the scene he lays down all the production audio tracks and then later on auditions the isos separately to pick the best ones in order to build the best-sounding scene. On Mission, his Avid timeline had six mono tracks for dialogue, four monos and six stereos for effects, and three stereos for music.

The team worked closely with the sound supervisor, sending him sequences so that he could send back stems of sound effects that Eddie's assistants would lay back into his Avid timeline. That way, the director could get used to hearing the sequences with effects, enabling him to give notes on sound design. It also meant that the screenings had better sound mixes.

## Communication

Several members of Eddie's team have worked with him over many films, and they've established an organizational methodology that enables him to work quickly—and for any of the team to understand exactly what the others have done. It's one of the unconventional aspects of working with such an experienced and finely-tuned team—the hierarchy they have is less defined and more horizontal than in many cutting rooms, and Eddie feels comfortable making requests to anyone on the team and knowing that it will be handled appropriately.

Communication, therefore, is one of the keys to their success. The codebook acts as the ultimate repository of all workflow information, a centralized database for all post departments.

They also used the codebook to track any requests Eddie might have had for insert shots for a particular scene. The assistants would review Eddie's Avid timeline, where he'd insert small captions requesting, for example, a close-up on a gadget from production. They could add that to the codebook with a thumbnail of the shot so that it could, in turn, be distributed to the set and tracked when it was received back to the cutting room. The team also used shared Google docs, iMessage groups, and Apple shared reminders (in the VFX group) to tick off tasks and initial their work.

Nothing, however, could replace actual face-to-face communication. The two first assistants shared a room so they could easily talk to one another (and the post supervisor) to ensure that nothing slipped through the cracks. That way, if they needed to pivot or punt, they were able to coordinate their efforts in the most sensible and efficient way possible.

## The post-shoot workflow

Because Mission had such a heavy VFX load, the post-shoot workflow commenced earlier than on most movies, with shots going to Double Negative (as 4K EXR files) beginning only a few weeks into the shoot. Later on, the other post-production components such as sound, music, the stereo conversion, and DI kicked in, along with trailers, screenings, and marketing requests.

Later on, a flurry of turnovers ensued as VFX shots were returned to the cutting room to be dropped in, music and sound effects were composed and completed, the "reasonably locked" sequences were sent to Prime Focus so they could begin the laborious process of stereo conversion and QC, and Molinare color graded the DI. Fluent Image was responsible for distributing the 4K files and communicating with the cutting room to ensure proper tracking.

Meanwhile, test screenings were conducted roughly every two weeks, and audience feedback had to be tracked and the cut updated accordingly.

As ingenious as their long-distance workflow was, when the final stages of post were in progress, it was extremely helpful that the main cutting room was located in Soho, walking distance from Double Negative, Molinare, and De Lane Lea, where they did the sound post. That made it easier for Chris McQuarrie, Eddie, and any of the other post crew to be in the various rooms to speak with the rest of artists involved.

The final deliverables were 4K DIs in 1.90:1 IMAX and 2.39:1 anamorphic, along with a 3-D (stereo) version. DCPs (digital cinema packages) are the common delivery medium.

## The bottom line

Tom Cruise went to extreme extremes to deliver a film that raised the bar on the action genre. The production and post-production crew likewise raised their games, creating and executing the ultimate cutting-edge workflow. And that workflow was built upon extreme preparation, organization, and communication.

But here's the bottom line: telling the best story is always the goal. If it isn't, it doesn't matter how expensive the gear is or how complex the workflow. Because the workflow should always be designed to help you tell your story the best way possible.

Independent Film – Searching

The recent film *Searching*, directed by Aneesh Chaganty and edited by Nick Johnson and Will Merrick, was a conceptual breakthrough. The story is told entirely through computer and handheld-device screens, which involved the creators' devising a workflow that reflected both the new ways in which we communicate—and make motion pictures.

After breaking down the multi-tiered creative process in a Frame.io Insider article, we were so intrigued by what is clearly a unique filmmaking process that we decided to drill down into the nuts and bolts of their equally unconventional technical workflow.

## Overview

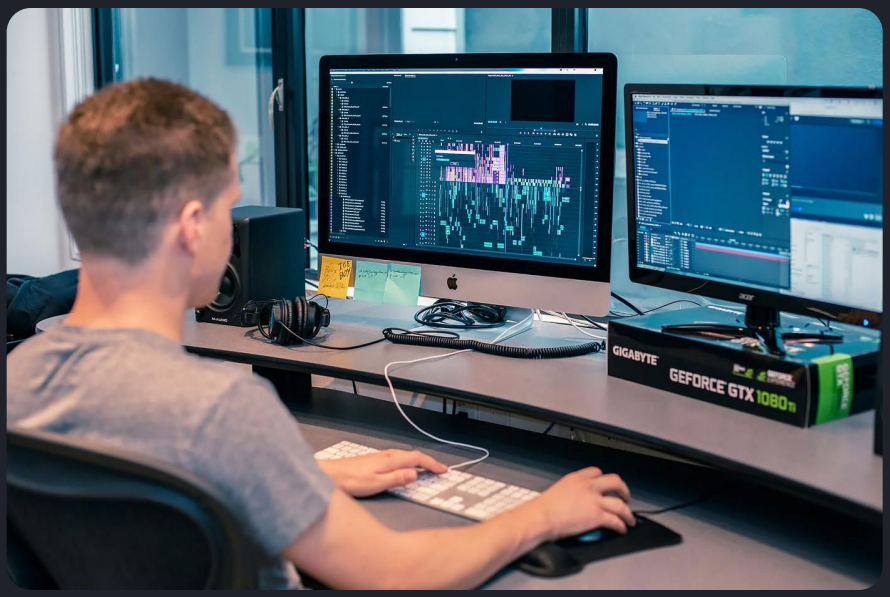
The production was divided into phases: "pre-post", in which the editors sat with the director and created and choreographed an animatic of the entire movie; "post," in which they edited the live action into the screens based on the animatic; "and post-post," in which they added all of the moving graphical elements and sound.

## The Cameras

In keeping with the film's central conceit, the cameras were mostly GoPros and iPhones, with Sony Z100s for news footage, and drones for some of the aerial shots. The live-action images were intended to be viewed through a digital device, so using those cameras lent authenticity to the film's aesthetic.

## Editorial Prep

The editors chose to cut in Premiere running on iMacs, knowing that they were going to go back and forth in Illustrator and After Effects to create (and integrate) the screen graphics. The ease of moving elements within the Adobe creative suite made it a logical choice.




As a result of using mainly pro-sumer (or even consumer) acquisition, with many different frame rates and sizes, the editors had to transcode all the media from h264 to ProRes 422. To complicate matters further, the screen records they

captured on the set were at 30 fps, so they had mixed frame rates even within the timeline.

In spite of the prosumer video capture, audio was recorded separately on the set to a professional external recorder, and the DIT sent the footage to editorial already synced, using an automated syncing application and then checking it by hand against the slates. Both editors noted that because much of the footage was shot on iPhones there were occasional shifts in frame rate that resulted in audio drift, so that even in the editorial phase they had to adjust the sync by eye.

Everything was recorded in Rec. 709 color space, with no baked-in LUTs, although they did employ some of the Premiere LUTs during the editorial process on adjustment layers.

## Pre-post and post

The movie was divided into 26 separate sections, each of which was assigned a letter of the alphabet. As the sections were completed, they were combined to form reels. Both editors worked on all of the sections (and reels), albeit at different times. Because Premiere's Team Project function was still in beta, they collaborated in a more manual way.

They each ran their own projects and initialed them, and then would import one another's sequences. When they took a sequence in, they'd immediately rename it and initial it, including the time—down to the minute—as part of the naming convention, a trick Nick learned from his time at Buzzfeed.

Everything existed in two project files, one for each editor. Communication between them was essential, and being in adjoining rooms with a door they could keep open made it

much simpler for them to just talk to each other to avoid confusion.

There were backups of all the production footage on a shared third-party RAID server from QOS for all the media, and they frequently backed up their project files onto a 16TB drive along with the screen graphics, which constituted the bulk of the footage. Because there were so many graphical elements, they created separate folders for such assets as mouse cursors, pointers, and desktops.

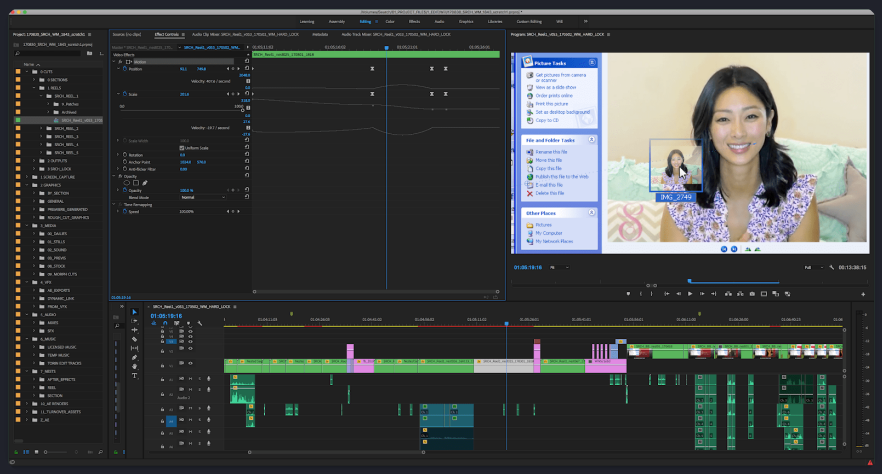
In terms of overall editorial process, even before the first frame of video was shot, the two editors worked for an incredibly brisk seven weeks with the director to create a full-length animatic template of the entire film from the initial treatment. In fact, the shooting script was created *from* the animatic. From day one of the animatic process, they used Frame.io extensively for reviews with the director and producers. With only seven weeks to essentially pre-edit the entire movie, fast turnaround and accurate feedback were critical—and remained so throughout the entire production.

Once shooting began, the process became more traditional, with the editors cutting in dailies over the course of the fourteen-day shoot. Because they had pre-timed placeholders in the animatic, it was more a matter of replacing the placeholders with the production footage and adjusting the timing. But that's where the traditional part of the workflow began and ended.

After shooting wrapped, Nick and Will had approximately two months to complete the director's cut, and about another month to complete graphics and the post-post processes.

## Graphics and templates

Nick and Will worked extensively in Illustrator to create the graphics and then linked them into the Premiere timeline. Initially, they looked blurry because Premiere doesn't rasterize them through the nests. However, once the Illustrator files were imported to After Effects, all they had to do was turn on a feature called Continuous Raster and the graphics were immediately up-resed.



Neon Robotics handled the After Effects work of creating graphic templates such as the Chrome top bar, Gmail inboxes, and the iMessage template. The editors were able to Slack the team at Neon to request those kinds of more complicated graphics and then were able to plug them in and modify them by adding the bubbles and text. Neon also created the news graphics and handled some of the invisible effects, such as adding trees to an aerial drone shot, working off of DPX sequences of the original footage from the editors.

Titles were also created in After Effects and the producers used Endcrawl for the credits.

## Sound

The sound was done "fairly" traditionally, according to Nick and Will. They worked with a temp soundtrack during the

editorial process, and the sound designers began work after the main picture was locked, using AAF files from Premiere. The trickiest part was that the editors were still working on the graphics–timings for mouse movements and clicks, and typing keystrokes and whatnot–so they had to go back and forth with the sound design to make sure that the visuals and sound lined up precisely. Because the audience is so accustomed to the way the sounds of keyboards and clicks coincide with hand motions, even a small amount of drift was distracting.

## Color correction

Working with a colorist who was experienced with non-traditional workflows, they were able to handle the majority of the color correction in After Effects using Color Finesse. Nick and Will built a temp version of every layer of the film in Premiere, which needed to be fully locked before exporting to After Effects (they emphasized to producers and the director that once the sections went into After Effects they wouldn't be able to go back into Premiere for editorial changes again without rebuilding the sections). When they had final approval, the editors took the locked sections and moved them over to After Effects using Dynamic Link as the handover tool, and completed the color correction, essentially creating an After Effects clone.

At that point, they rendered DPX sequences out of After Effects in order to go to Resolve for pretty simple vignetting and white balance correction. The editors added the cut points in for the colorist, and then after he completed the final color pass they rendered out reels in ProRes 4444 from Resolve.

## Archiving

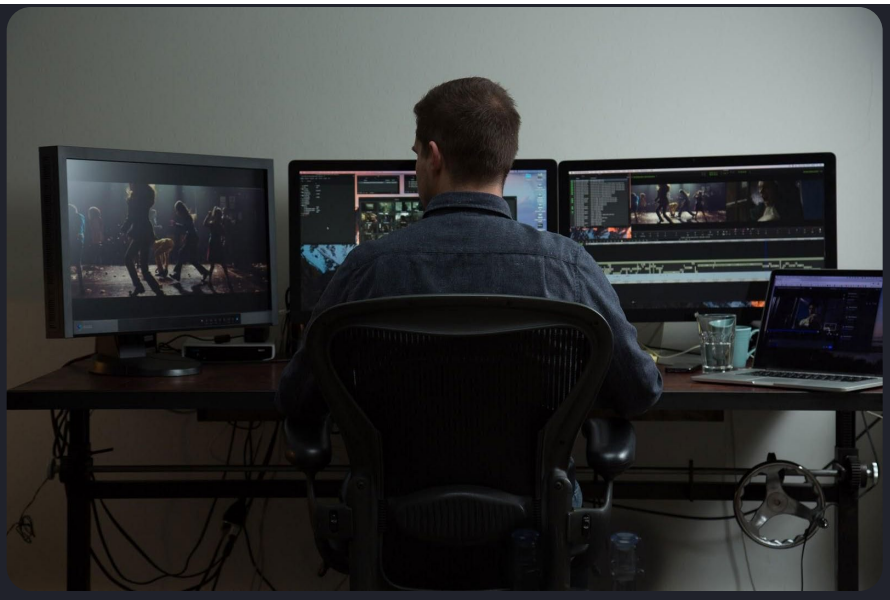
The final was archived to LTO tape and turned over to Sony.

## The bottom line

With a production budget of approximately $1 million and a worldwide gross of approximately $70 million, *Searching* was a success by all accounts. While most theatrically released films are shot on expensive cameras, cut on Avids, and go through the more traditional DI process, in this case the Adobe Creative Suite helped the filmmakers create a movie–and a workflow–that will likely inspire others to explore the kinds of unconventional processes that make moviemaking more accessible–and more exciting.

Broadcast Ad – Sunoco

The slick, high-end ads you see in movie theaters or during the Superbowl are some of the highest cost-per-minute productions on the planet. Typically shot by the most accomplished directors working in film and television, they're also cut by the best commercial editors at boutique editorial houses such as The Assembly Rooms in London. David Stevens, whose work on the Harry Styles video "Kiwi" was featured in the Frame.io Insider, gives us a detailed account of his workflow and why it works.

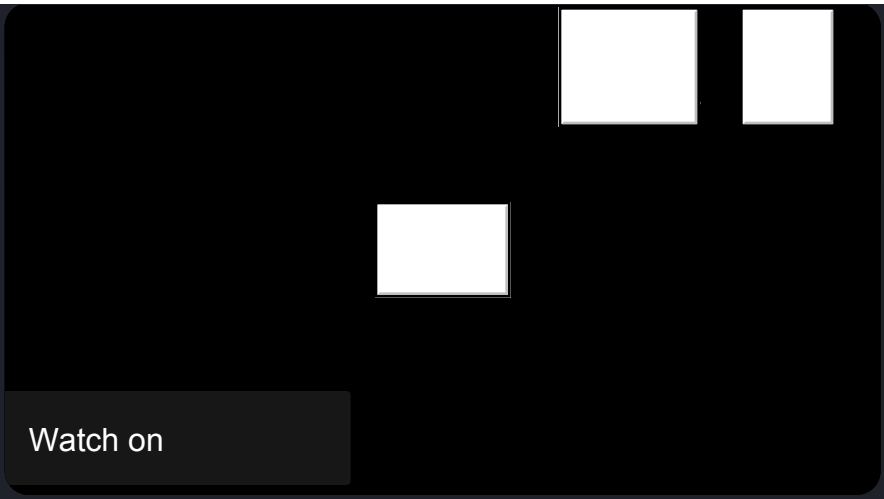


## The Overview

It's rare that an editorial house has a say in the cameras that are used, but David is sometimes consulted prior to shooting if there's a particular effect the director is trying to achieve, such as extreme wide angles or very high speed photography for an extreme slow-motion effect. Typically, however, for high-end commercials, Arri Alexas or similar premium camera equipment is the norm, giving the director the greatest creative latitude. There's even a growing trend among commercial directors to shoot on 35mm film.

Occasionally, David may attend the shoot. One example is this Sunoco spot helmed by Chris Cairns of Partizan, which was dependent on the careful choreography of the sounds of the mechanical motorsport elements that were being shot live in the studio. David was drip-fed visuals from the DIT after each take so he could experiment with edit techniques and let the director know if he had adequate coverage before moving on.

Sunoco "Symphony" featu



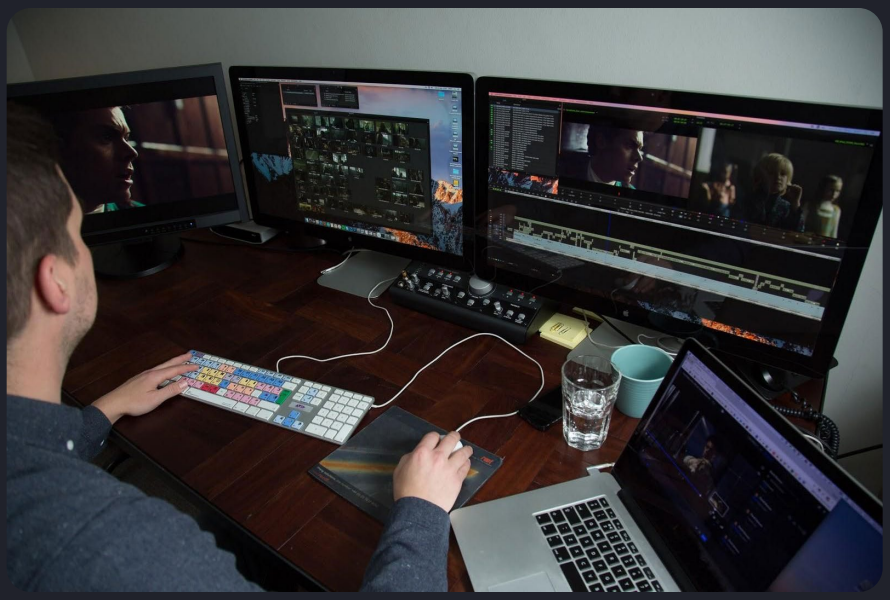

It's a workflow that's more typical on a feature film than on a commercial, but because many high-end commercials are, in a sense, mini features with similarly high budgets (combined with far tighter schedules), it's not uncommon for an editor (or visual effects supervisor/team) to be on set. As David notes, sometimes it's beneficial for an editor to be there to understand how the director intends for the piece to cut together. Conversely, it can be more beneficial for the editor to see the footage without being influenced by what happened on the set in order to be more objective about what's in the footage itself, functioning as an expert pair of fresh eyes.

## Editorial Prep

The Assembly Rooms NLE of choice is Avid. However the footage is shot or formatted, it's always transcoded to DNxHD 36, which provides adequate resolution for the offline while reducing the data load to manageable file sizes. This is especially important at an editorial house in which many editors are working on multiple projects concurrently. It also enables an editor to put the media onto a single drive in the event they need to work on the set, at a different location, or remotely. DNxHD also improves the speed and stability of the computing platform and reduces (or eliminates) render time–

a big consideration for editors who are working on the frequently very limited schedules commercials have.

David currently works on Avid version 8.10 on a 2017 iMac Pro 3.2GHz Intel Xeon W with 32gb memory and Radeon Pro Vega 56 8176 MB graphics card running on OS 10.13.5. The Assembly Rooms use a variety of in/out boxes such as the AJA IO Express to allow an on-desk Eizo monitor and a large client-facing Sony monitor. For sound, they use an on-desk amplifier and Bose speakers or self-amplified Rocket G3 speakers.



Normally, the on-set DIT will transcode the files immediately after shooting, and a drive is sent to The Assembly Rooms either at the end of each shoot day or at the end of the shoot (depending on schedules and logistics). Sometimes the files are fed via upload, particularly if the shoot is in a faraway location or the director is unable to be physically present for the edit. If there is no DIT on the set to perform the transcodes, the editorial assistant can handle them—though it slows the process down.

Once the files arrive at The Assembly Rooms, they transfer the transcodes to the Avid Unity server and ingest to the Avid project. This way, both the assistant and the editor can work in

tandem easily. The transcodes are stored on the Unity server, which backs up nightly. They also hold the hard drive that contains the transcodes from the production in their drive library, and the production company keeps a copy of the master drive at a different location in case of any mishaps.

For the Avid media, they use an Avid Unity system that creates separate partitions for each project. Each partition then follows the standard Avid media directory layout, which allows multiple machines to access the same media at the same time.

They also have an IMPORT/EXPORT shared drive, where all scripts, camera reports, and storyboards are saved for reference, along with any music or graphics that may need to be provided to the sound or the final post facility. They also keep it in the event there is ever an issue with the Avid media and they need to rebuild the project.

If sound syncing is required, David prefers that the production use a digi-slate so that the sync process can be automated—either as part of the transcoding process or within Avid as part of the project setup. Clapper boards also work well for syncing. Minimally, in situations of one-off or unpredictable events, or run-and-gun shooting, even a post-take clap helps in the syncing process.

Note that because many of the commercials and music videos David edits require vocal performances or music that needs to be played back on set as a reference track, he'll provide the track with a visual timecode so it can be played back from an iPad that's held in front of the camera, much like a digi-slate. This aids in syncing the performance takes.

## Organization

For short-form projects such as commercials and music videos, The Assembly Rooms has created a system designed to maximize creative time and future-proof their work. They maintain the philosophy that the shared organizational approach should enable any editor to pick up another editor's project and be able to work on it without a problem. It's important, for example, in cases where a client might need a revision and the original editor is already booked on another project or is otherwise unavailable.

The dailies are organized into "slates" dictated by the information on the clapper board. If there are no clapper boards, then the footage is divided by setup and they create numbers to organize each setup.

Elsewhere in the project, they keep a separate folder for all audio, with separate bins for sync sound, Foley, sound effects, voice-overs, and music.

All elements such as graphics, logos, animation, and work-in-progress visual effects, are kept in an Avid IMPORT folder. They create an EXPORT folder for all EDL, AAF, and viewing copy exports.

They also create a MISC or ASSIST folder for all work-in-progress Avid bins such as sync bins and rough composites.

All of this organization to keep the project tidy allows the editor to move through the material as quickly and efficiently as possible. On jobs of this nature, it's common for the editor to have very limited time with the director, and this level of organization allows David to maximize that time.

## The Editorial Process

Once David's assistant has set up the Avid project, David watches "absolutely" everything. He'll then perform a first pass of wide selects to get a feel for the footage, pulling out the best moments and removing anything unusable. He, like other editors working at this level, feels it's important for him to be as intimate with the material as the director–if not even more so.

Depending on the quantity of footage, he may then do a second pass at selects to further distill the material. In cases where there are numerous takes of particular action and the director has indicated his or her selects, David may string together a very rough assembly just to get a sense of the whole–although that approach is most relevant to carefully boarded commercials with a distinct story or script.

But in the case of a spot like the Sunoco piece, which is less dependent on a scripted story and more on the flow of the action, David approaches his first assembly from a completely creative standpoint, relying on his instincts to guide his choices. It's an essential step, resulting in a foundation upon which subsequent discussions with the director or client are based. It's also why having the material organized is important –he doesn't want the creative flow to be interrupted by having to spend extra time sorting through alternate selects.

It's worth noting that it's not uncommon for commercial editors to create their first-pass edit without clients being physically present. On this spot, David already knew what the director was going for, having been at the shoot. In other instances, he'll likely have had a detailed conversation with the director about their vision, even (if time and proximity allow) going through the initial selects together.

## The Review Process

Reviews can happen either with the director and clients present or remotely. David finds the time with the director to be "deeply rewarding" in terms of the creative collaboration sparking new and better approaches.

If, however, he must present remotely, he relies on Frame.io. He and the director will use the commenting and markup tools to collaborate together as they go through various drafts. Once they have a cut they're satisfied with, David will use Frame.io's presentation page for client reviews, which presents the cut as more of a finished piece to be judged as a whole.

In the case of presentations with clients there in the room, David takes extra care to make the room as comfortable as possible—ensuring the sound is good, there's no screen glare, no distracting noise. He'll play the cut through twice, letting the clients absorb it before proceeding to discussion, which is equally organized: they'll first discuss the piece as a whole and then address individual points.

## Outside Vendors

Boutique editorial houses like The Assembly Rooms tend to focus exclusively on offlining, handing off to a post house for online conforming. David provides an EDL and visual AAF to the post house so that the cut can be relinked and conformed using the original full-resolution master files or camera negative. Because the color grading is an important step, it's typical for the cameras to record in log if not raw, as that gives the colorist the most latitude to create the final look.

David may attend the color grading and/or online sessions in order to make sure that the handoff goes smoothly and that all the creative decisions (such as combining multiple plates, speed changes, and flopping or reversing shots, etc.) that

have been made are communicated to the colorist and online editor.

Depending on how much time is allowed in the offline, David may do a fair amount of sound work. Often there will be placeholder music, a mix of all the channels recorded on set, a scratch VO, and sound effects that an assistant or the editor may pull.

In the case of the Sunoco spot, David was constantly adding new shots and sounds to mix up the mechanical "music" they were creating. He worked closely with Mike Bamford at String & Tins, and likened the experience to writing a song together.

More commonly, however, the sound design and mix will be done elsewhere, either at an audio or post house, for which David provides an audio AAF. Once the final audio has been completed, it will be laid back to the color-corrected and conformed master.

Visual effects are similarly handled by either a post house or specialized visual effects vendor, although David may do rough compositing using the Boris FX package along with Avid matte keying. It's not atypical for there to be some back and forth with the VFX vendor–David might have locked sequences or select shots that he can feed to them so they can get started, and as they complete their work they can hand it back to be dropped into the offline.

In the end, the post house is responsible for adding titles, creating versions (for foreign languages or customized markets, for example), and delivering the finals to the specifications of the client.

## Archiving

The Assembly Rooms keeps active projects on the ISIS data server for 6-12 months, in the event of revisions. After that, the offline materials are loaded onto individual hard drives that are logged with their contents for easy searching, and stored in a fire-safe vault for between two and five years.

Avid projects are kept indefinitely because they are small and easy to store, with the thought that if anything else is requested beyond what's part of the standard archiving process, the Avid project could be reloaded and re-linked from the master files (which are kept with the production company or client). Those sorts of requests are rare, but when they do occur, a little extra care in archiving and storing can make all the difference in being able to accommodate them.

## The Bottom Line

High-end commercials are expensive to make and often come with short schedules. Editors who work on those projects have to be incredibly well organized so their time is spent in pursuit of making the project the best it can be. It's also what allows them to give the client the kind of highly-personalized, creatively-focused experience they expect. And, it's what makes for repeat business and enduring professional collaborations.

## YouTube Star – Film Riot

When you're trying to produce pro-quality content on an indy-film budget, you have to work smart. No one knows that better than Ryan Connolly, independent filmmaker and founder of Film Riot, and in his YouTube videos and podcasts he's happy to share his knowledge with filmmakers of many

levels.

He's a repository of tips and tricks that can help you up your game while saving money and time. Here's a look at the streamlined studio workflow Film Riot uses to produce their twice-weekly episodes.

## Shooting

The main cameras used for their how-to videos are the Canon C100 and C300 Mark II, and the Sony A7S II. The C100 is used primarily for their locked-down green-screen shoots; the C300 serves as the traveling camera for shooting their sketches; and the A7S is locked down at Ryan's desk as he does his Q&As or software walk-thrus.

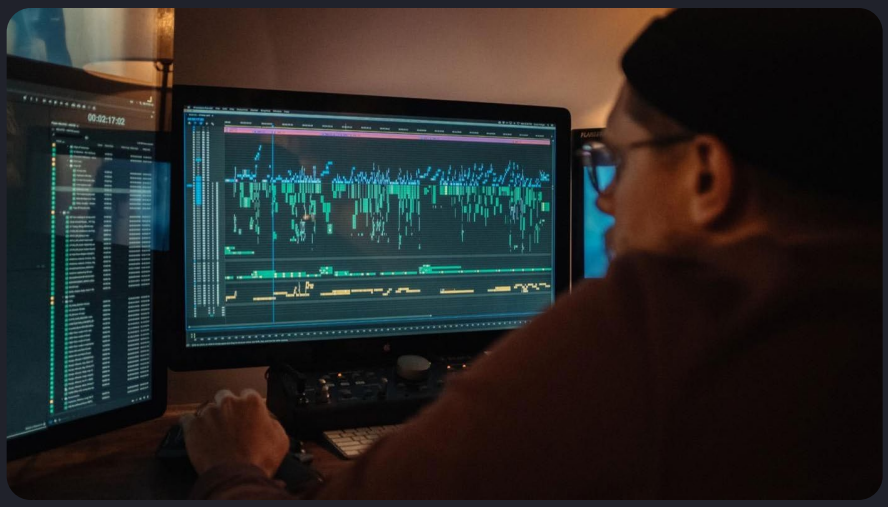
They always shoot at 24 fps and almost never record in log (color grading is handled with their own collection of LUTs). And with turnarounds ranging from a day to three days, they also rarely shoot in 4K.

Film Riot uses in-camera audio, further reducing turnaround

time by eliminating the need for post syncing and the cost of recording audio separately.

## Editorial

Because speed and ease is of paramount importance, the Film Riot crew uses raw camera files without transcoding, which work well in Premiere with XAVC-S as the primary codec. In fact, when Ryan's finished doing his on-camera shoot, he just pops the card out of the camera that's mounted on his desk and into his desktop system, the HP Z40, which he describes as "a beast of a system."



Josh Connolly, Ryan's brother, is his Film Riot partner, and he handles offloading and backing up the files to a custom-built NAS setup. Nothing is stored exclusively on the individual systems. Working from the NAS, both of them can access everything from files to assets to plugins.

Josh also sets up the folders per episode for both he and Ryan to use. Once upon a time, they created an episode template, but currently they find that the individual episodes are so different that the template has become less useful. The basic structure is an Adobe folder for all of the Premiere and

After Effects files, an audio folder that branches out to music and sound effects, a folder that contains the sketch material, and then additional folders (as needed) for any screen captures or VFX.

Both Ryan and Josh edit–Ryan handles the sketches, while Josh assembles the episodes (on an Alienware system that is less tricked out than the HP), with Ryan stepping in to do the final color grading and audio sweetening.

While Ryan and Josh effectively are the core of Film Riot, they employ an off-site After Effects artist from England who handles graphics and VFX, and have another part-time artist who's based in Ohio. Film Riot also has an online store, and they're collaborating with editor Lucas Harger, who edited Ryan's recent short film *BALLiSTIC (link to the MIF piece?),* on a store-related project. To stay organized, they use Google Drive and Dropbox to upload assets, and Frame.io for notes, reviews, and sharing versions.

Ryan does his own color grading for the Film Riot videos, using Premiere and Lumetri. He likes to add a little bit of grain at the end using FilmConvert, and sometimes uses Magic Bullet Colorista or Magic Bullet Looks, working on the adjustment layer rather than the individual clips.

Sound is handled in Premiere if it's very simple (again, to save extra steps) but if Ryan's working on a sketch piece he'll do a dynamic link to Audition and send a mixed-down version of the file back to Premiere. For more complex sweetening he might also use Ableton Live. And Soundly is Ryan's game-changing sound effects source—for a monthly fee you get access to thousands of sound effects that are easily searchable, simple to manipulate, and can be dragged directly into Premiere. The ease and speed of Soundly makes it one of Ryan's go-to pieces of software.

## Delivery and Archiving

Because they're delivering the videos to Facebook and YouTube, delivery is straightforward, with a stock h.264 upload out of Google Drive, Dropbox, or Frame.io, depending on what's most convenient at the time. Note that the short films are a different story, uploaded to Vimeo at ProRes.

As for archiving, Film Riot keeps everything stored on their 60 TB server, with the caveat that they have amassed a quantity of content that will make it necessary for them to find a better long-term solution soon. Of course, when you're not storing client's materials, it makes finding a solution just a little less pressing.

## The Bottom Line

When you're an independent filmmaker, you always need to get as close to doing the best work as quickly and inexpensively as possible. It's a big ask, but Film Riot has it pretty well figured out.

Weddings – Cinemacake

The wedding industry is big business. Each year, approximately 2.5 million weddings take place in the U.S., at a cost of approximately $72 billion.

Image © mkPhotography

CinemaCake, an award-winning video production company in Philadelphia, creates videos for commercial, web, and corporate clients. But a large part of their business comes from creating meaningful films documenting once-in-a-lifetime events. The extra pressure of having to get it right the first time makes them ultimate pros in their sector of the industry. Ben Blevins, post-production manager of CinemaCake, takes us through their typical wedding video

workflow.

## The Shoot

The documentary "what you get is what you get" approach to capturing an event like a wedding demands excellent preparation and smooth execution.

The cameras have to be able to deliver high-quality images while being relatively small and unobtrusive. For that reason, CinemaCake owner Dave Williams chose the Canon 5D mark iii kit. It fits into small spaces and the color quality is exceptional.

They also use the Sony A7S with Canon lenses for its superior performance in low-light conditions and ability to zoom in live during a shot. To capture the more action-oriented parts of the wedding, they have four GoPros, as well as a DJI drone for aerial or hard-to-reach angles.

They shoot at 24 fps at 1080p, which means that they can use the raw camera data without transcoding, and because the

images aren't as crisp as 4K they have less of a need to use beauty or diffusion filters.

Each shoot uses at least four cameras, with three shooters who are mobile and one locked-off camera to take the wide-angle or tight angle, depending on the circumstances.

Audio is captured separately. They'll set up a mixer to monitor levels throughout the shoot. The Zoom H5 and Roland R-26 portable recorders are used for the main sound recording, along with a Tascam DR10 to record the Lavalier mic feeds from the groom and officiant.

## Editorial Prep

CinemaCake uses flash cards, SD, and micro SD cards, and after the shoot either Ben, or a contractor who is very familiar with their process, makes sure that each card is specifically labeled and numbered so that when the download occurs they know which card to go back to in the case of an incomplete transfer.

The camera cards are also labeled with the shooter's name, which makes it easier for Dave to review people's work and offer specific suggestions for future projects.

The audio is labeled with each source so that the editors don't have to listen to entire audio clips in order to determine which source it's coming from. For example, the audio from one card might be labeled as "direct feed," which indicates to the editor that it's the best source, while the other files are labeled to indicate "ambient" or "Lavalier" audio sources.

Ben noted that because the Sony A7S has a less detailed internal file-naming convention, if they happen to use multiple

A7S cameras on a shoot they use a tool called ReNamer to add extra detail so they can identify which footage came from which camera.

The raw camera files are offloaded to Drobo model 5D and 5N shells, which are basically five hard drives that fit together and work as a RAID. When the files are copied over to the Drobo, it automatically creates a redundant backup of all the data. And then, within a few days' time, the data is also transferred to a Drobo that's kept offsite for further protection.

No transcoding is necessary. CinemaCake's editors use either Premiere or Final Cut Pro X, and the codecs they use are dependent on the system. The only exception is possibly for time lapse or footage shot at higher or lower frame rates. Which NLE the editors use is strictly based on personal preference; for example, FCP X allows editors to change names or add labels in Finder within OSX and has a few other features that integrate especially well with the Mac OS.

There are six workstations, with one full-time staff editor (the rest are vetted independent contractors) and all work on 27-inch iMacs. They're equipped with HighPoint Technology's RocketStor docks into which the editors can plug their assigned production hard drives so that each can work locally on their particular project. It's an inexpensive, flexible, and practical solution.

Audio is synced using PluralEyes, although CinemaCake has observed that if more than two or three cameras have been used, PluralEyes can sometimes result in slippage or drift. Ben isn't a full-time hands-on editor, but he's found that Premiere's syncing capabilities can be more accurate. Audio is generally handled in Premiere (or FCP X) but sometimes the editors also use Audition if there's an inordinate amount of noise or clipping.

## Editing and approvals

The most common wedding deliverable is a 15-minute piece, which generally takes between ten and fourteen days to edit. Note that CinemaCake's editors have film school backgrounds and are both accomplished storytellers as well as technically adept operators.

Once a first cut is complete, editors upload the piece to Frame.io for Dave's input and approval. He'll make the first round of comments and let the editor take a second pass. After revisions are completed, Dave will give a second round of feedback (if necessary). Ben is responsible for QCing prior to delivery to the client. According to Ben, it's unusual for them to do more than a couple of passes, but if they need to do multiple versions, they use the Frame.io multiple version viewer so they can easily find the changes without having to view the entire piece. As each version is updated, they'll also change the status within Frame.io so the editor always knows when Dave or Ben have finished leaving feedback.

Soundstripe, Songfreedom, Musicbed and PremiumBeat (in that order) are the sites from which editors choose their music. After Effects and Photoshop are used for any visual effects or titling, and color grading is handled within Premiere (or FCP X).

## Delivery and archiving

After the video has been completely approved by the CinemaCake team, it's posted in a gallery on their website for the client to review on MediaZilla as a finished product. There may be minor adjustments, the first round of which CinemaCake will address at no extra charge, but typically they just have very happy clients.

When the client signs off on the MediaZilla version, CinemaCake can deliver the final, 1080p video on Blu-ray or a thumb drive, which many clients still like, while other clients will download their videos as a zip file from MediaZilla that contains a navigation menu like that of a DVD using Hightail as the delivery service. In rare cases, the client will elect to buy the raw footage as well, which is typically provided on a portable USB drive.

After the client receives their deliverables, CinemaCake pulls out the production drive and offloads the final ProRes version along with the project file and the music to two Seagate Barracuda drives. One drive stays in their offices and the second is kept offsite. The project is then deleted from the production drive.

Should the need to make changes after that occur, they would identify the Drobo that had the raw files, copy the files back onto the production drive, and then relink through Premiere or FCP X.

## The Bottom Line

When you're shooting someone's once-in-a-lifetime event, you have to make sure that you'll get the best possible coverage without disruptions or problems. Using small, unobtrusive cameras to capture different views and action along with multiple sound recorders and mics ensures that you'll have plenty of usable footage and creative options as you craft each client's unique story. Because while you can't control how an event will unfold, you'll need to have a workflow that's foolproof enough for you to capture the moments of unexpected beauty—or recover from the unexpected obstacles.